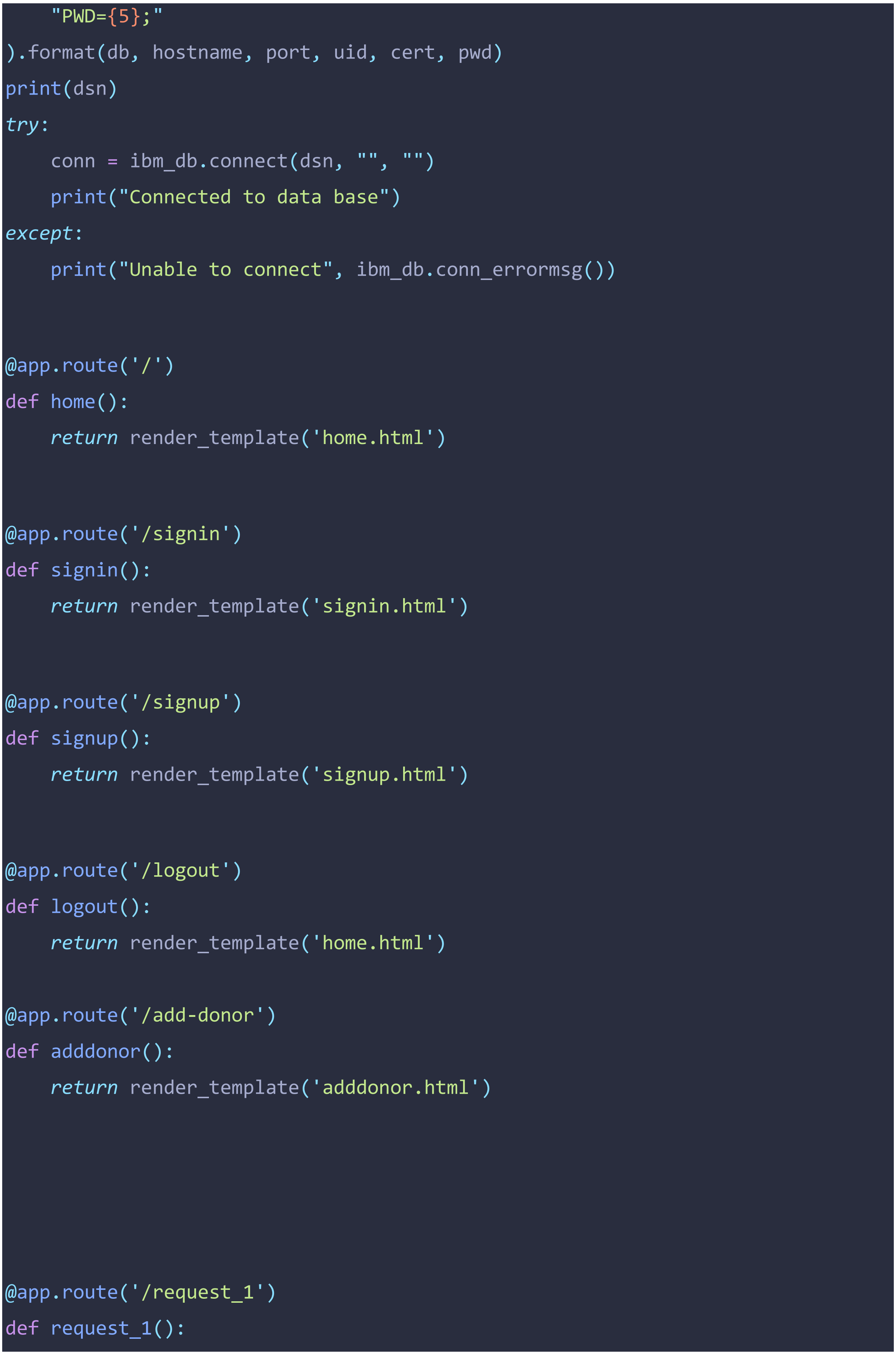
PROJECT DEVELOPMENT PHASE

SPRINT 4

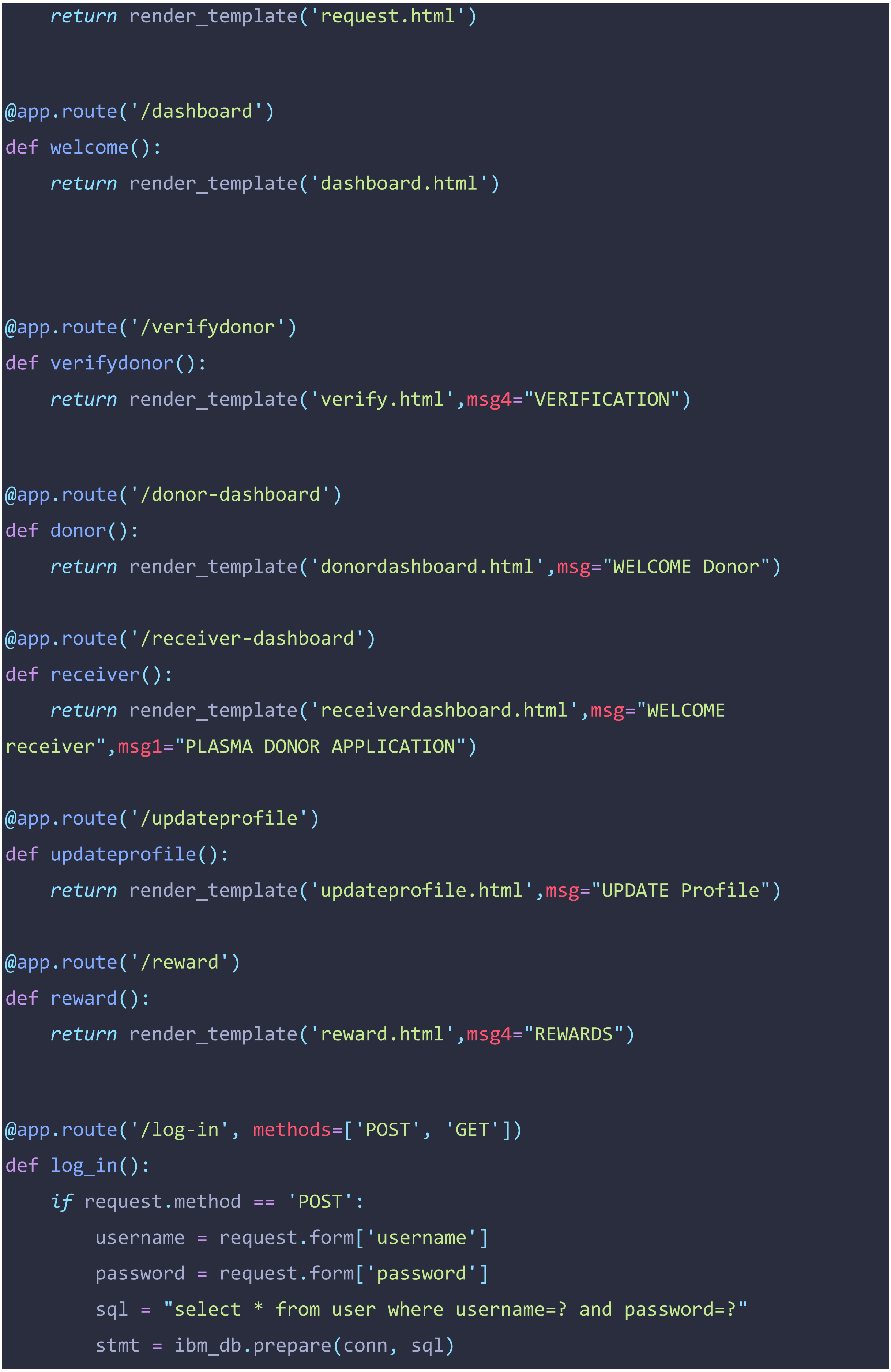
Team ID	PNT2022TMID48288
Project Name	Plasma Donor Application

FLASK APP:

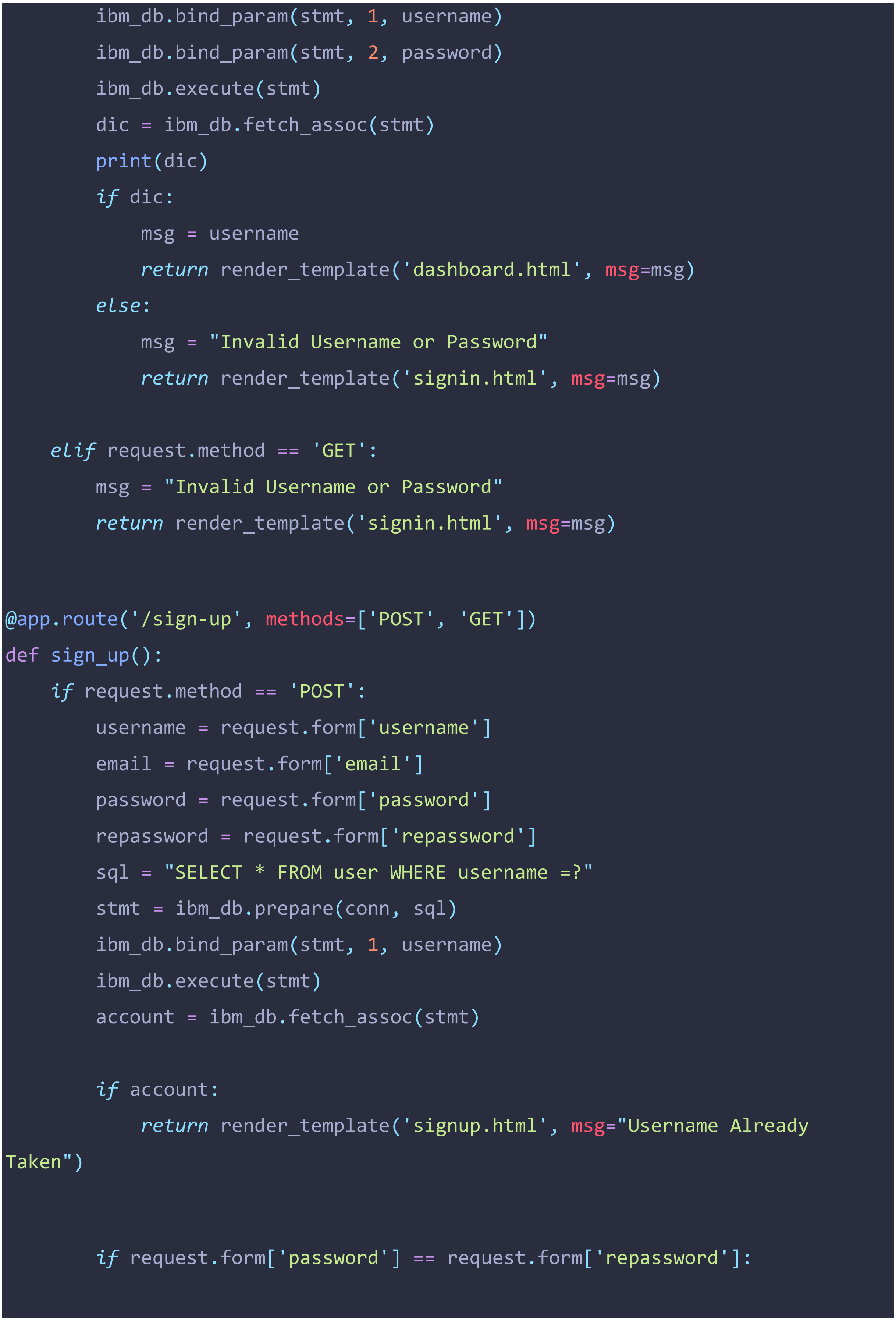
```
from turtle import st
from flask import Flask, render_template, request,redirect, url_for,
session,flash
from markupsafe import escape

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};"
```



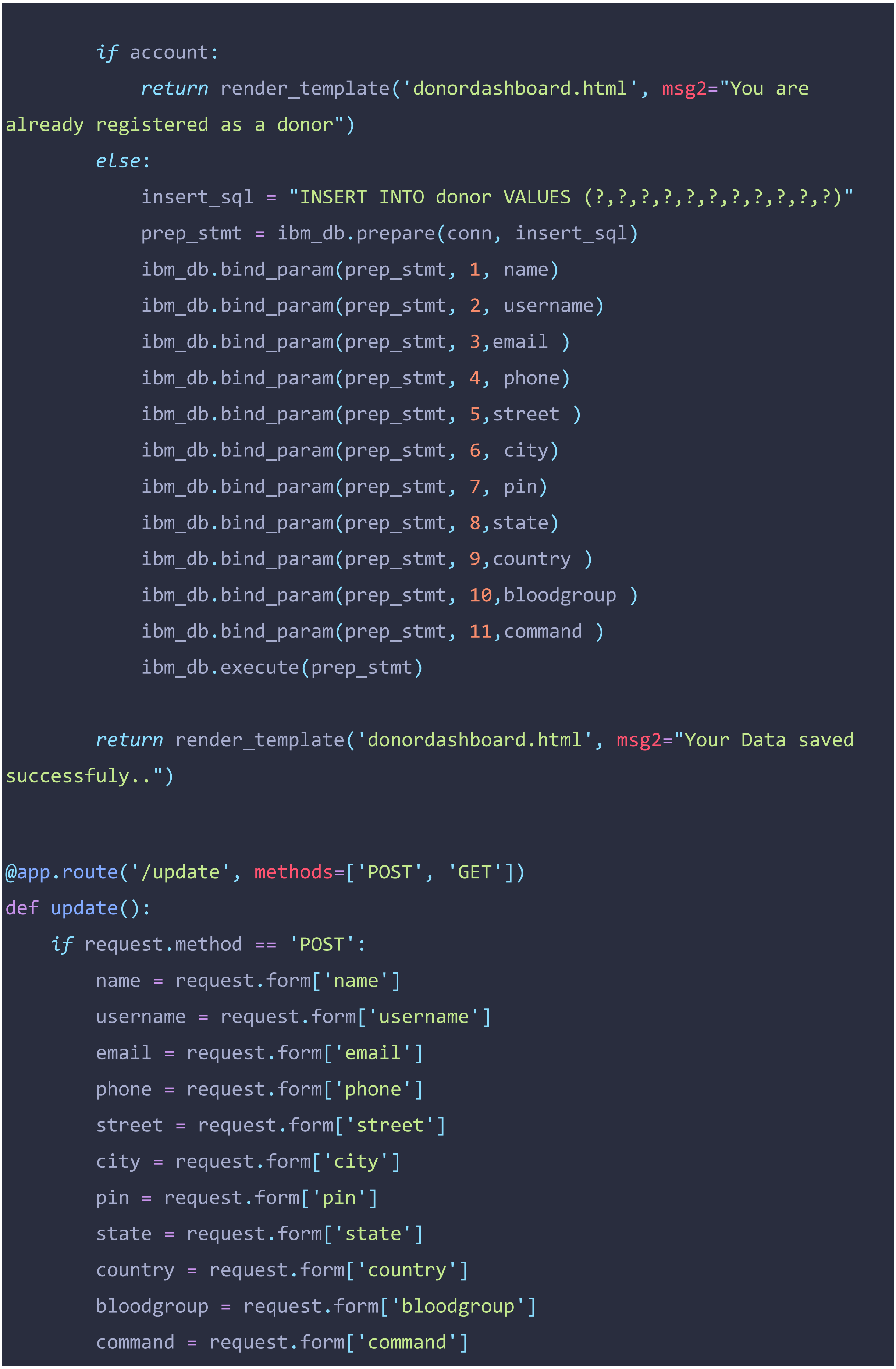
```
    "PWD={5};"
).format(db, hostname, port, uid, cert, pwd)
print(dsn)
try:
    conn = ibm_db.connect(dsn, "", "")
    print("Connected to data base")
except:
    print("Unable to connect", ibm_db.conn_errormsg())


@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():
```



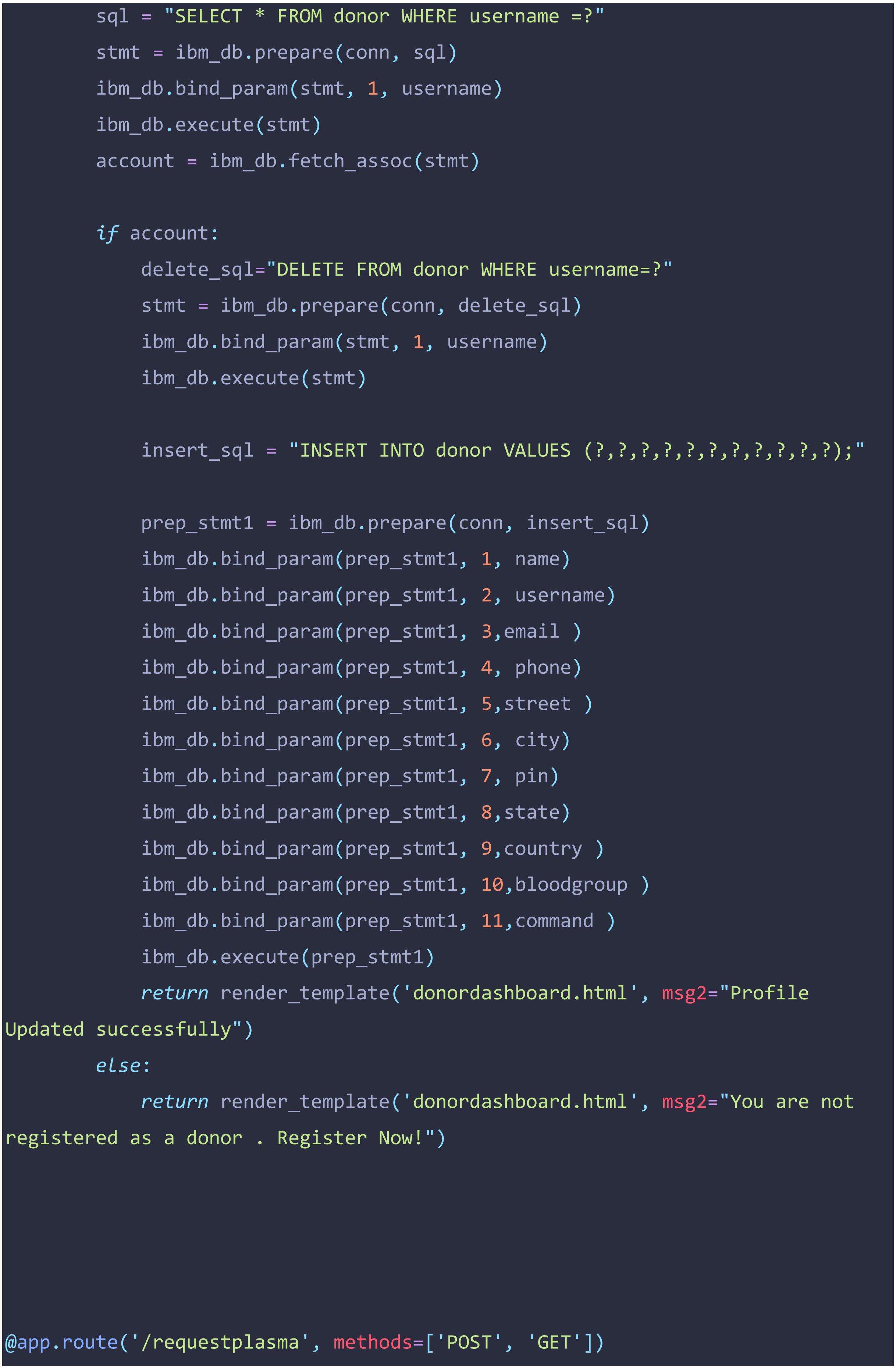
```
    return render_template('request.html')


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




@app.route('/verifydonor')
def verifydonor():
    return render_template('verify.html',msg4="VERIFICATION")


@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('/reward')
def reward():
    return render_template('reward.html',msg4="REWARDS")


@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(prep_stmt, 1, username)
            ibm_db.bind_param(prep_stmt, 2, email)
            ibm_db.bind_param(prep_stmt, 3, password)
            ibm_db.bind_param(prep_stmt, 4, repassword)
            ibm_db.execute(prep_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)
```

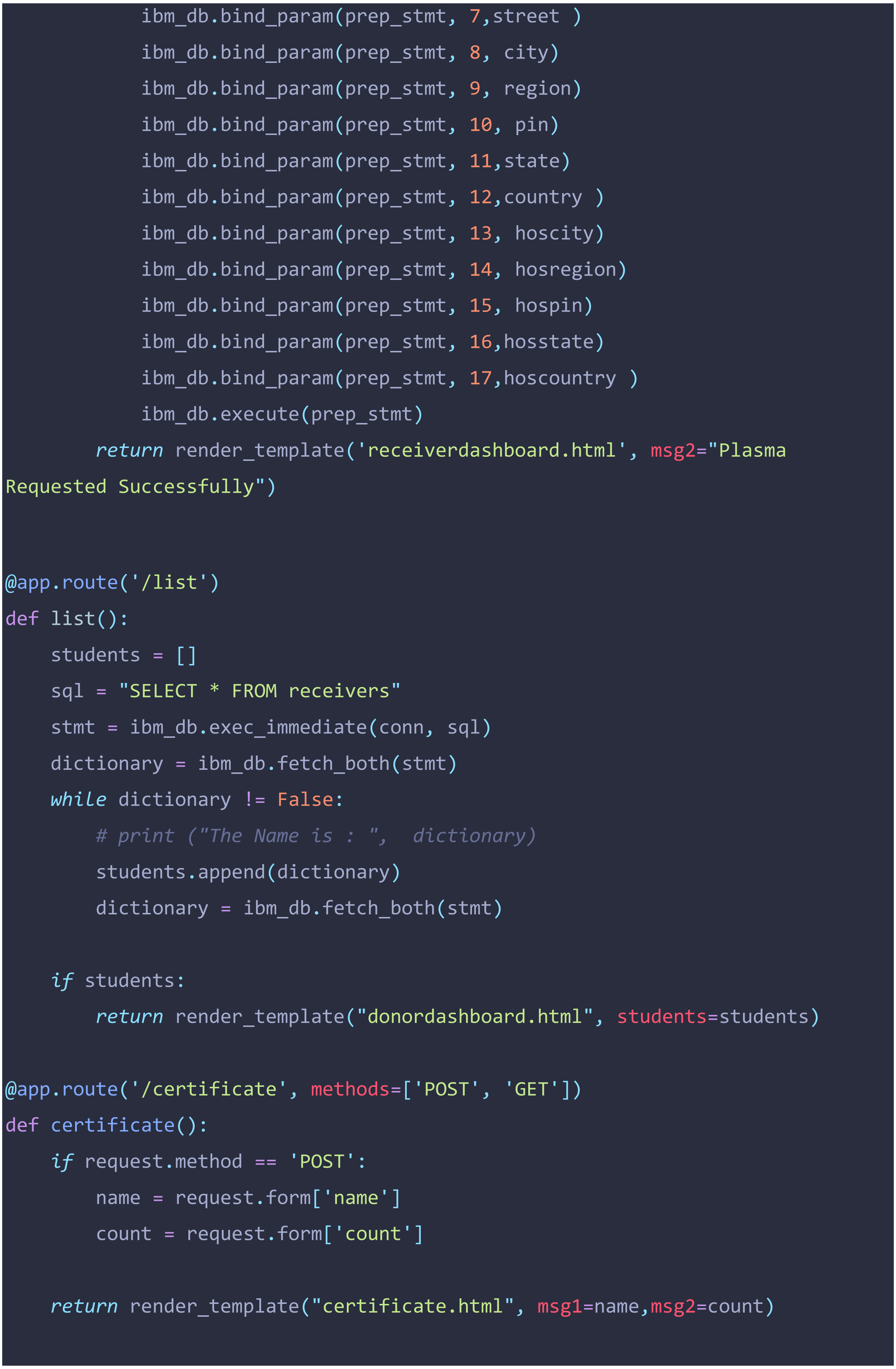


```
        if account:
            return render_template('donordashboard.html', msg2="You are already registered as a donor")
        else:
            insert_sql = "INSERT INTO donor VALUES (?,?,?,?,?,?,?,?,?,?,?)"
            prep_stmt = ibm_db.prepare(conn, insert_sql)
            ibm_db.bind_param(prep_stmt, 1, name)
            ibm_db.bind_param(prep_stmt, 2, username)
            ibm_db.bind_param(prep_stmt, 3,email )
            ibm_db.bind_param(prep_stmt, 4, phone)
            ibm_db.bind_param(prep_stmt, 5,street )
            ibm_db.bind_param(prep_stmt, 6, city)
            ibm_db.bind_param(prep_stmt, 7, pin)
            ibm_db.bind_param(prep_stmt, 8,state)
            ibm_db.bind_param(prep_stmt, 9,country )
            ibm_db.bind_param(prep_stmt, 10,bloodgroup )
            ibm_db.bind_param(prep_stmt, 11,command )
            ibm_db.execute(prep_stmt)

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


@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']
```

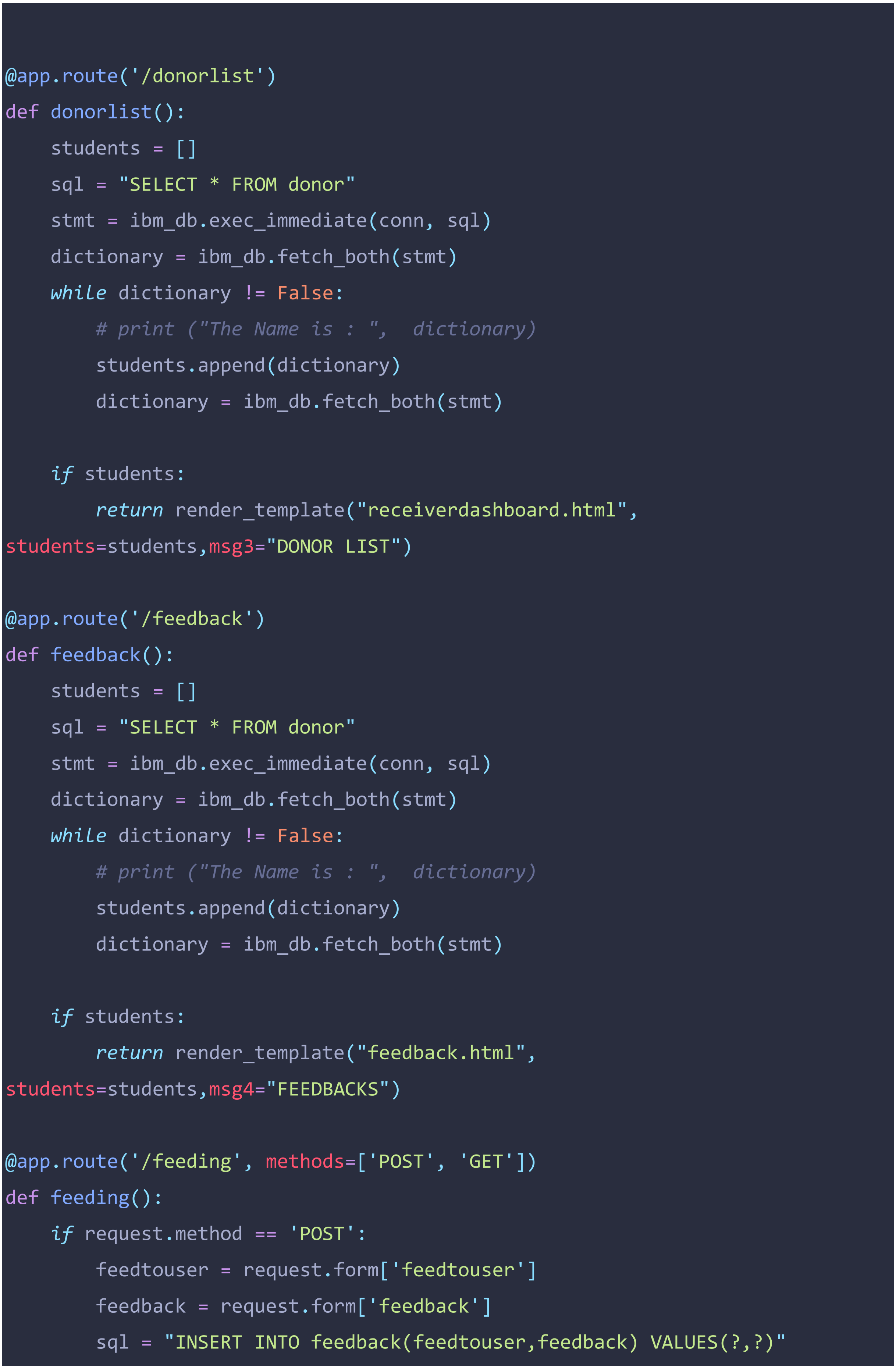


```
        sql = "SELECT * FROM donor WHERE username =?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt, 1, username)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)

        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(prep_stmt1, 1, name)
            ibm_db.bind_param(prep_stmt1, 2, username)
            ibm_db.bind_param(prep_stmt1, 3,email )
            ibm_db.bind_param(prep_stmt1, 4, phone)
            ibm_db.bind_param(prep_stmt1, 5,street )
            ibm_db.bind_param(prep_stmt1, 6, city)
            ibm_db.bind_param(prep_stmt1, 7, pin)
            ibm_db.bind_param(prep_stmt1, 8,state)
            ibm_db.bind_param(prep_stmt1, 9,country )
            ibm_db.bind_param(prep_stmt1, 10,bloodgroup )
            ibm_db.bind_param(prep_stmt1, 11,command )
            ibm_db.execute(prep_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']
        hoscity = request.form['hoscity']
        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(prep_stmt, 1, name)
            ibm_db.bind_param(prep_stmt, 2, username)
            ibm_db.bind_param(prep_stmt, 3, gender)
            ibm_db.bind_param(prep_stmt, 4, bloodgroup)
            ibm_db.bind_param(prep_stmt, 5,email )
            ibm_db.bind_param(prep_stmt, 6, phone)
```



```
            ibm_db.bind_param(prep_stmt, 7,street )
            ibm_db.bind_param(prep_stmt, 8, city)
            ibm_db.bind_param(prep_stmt, 9, region)
            ibm_db.bind_param(prep_stmt, 10, pin)
            ibm_db.bind_param(prep_stmt, 11,state)
            ibm_db.bind_param(prep_stmt, 12,country )
            ibm_db.bind_param(prep_stmt, 13, hoscity)
            ibm_db.bind_param(prep_stmt, 14, hosregion)
            ibm_db.bind_param(prep_stmt, 15, hospin)
            ibm_db.bind_param(prep_stmt, 16,hosstate)
            ibm_db.bind_param(prep_stmt, 17,hoscountry )
            ibm_db.execute(prep_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('/certificate', methods=['POST', 'GET'])
def certificate():
    if request.method == 'POST':
        name = request.form['name']
        count = request.form['count']

    return render_template("certificate.html", msg1=name,msg2=count)
```



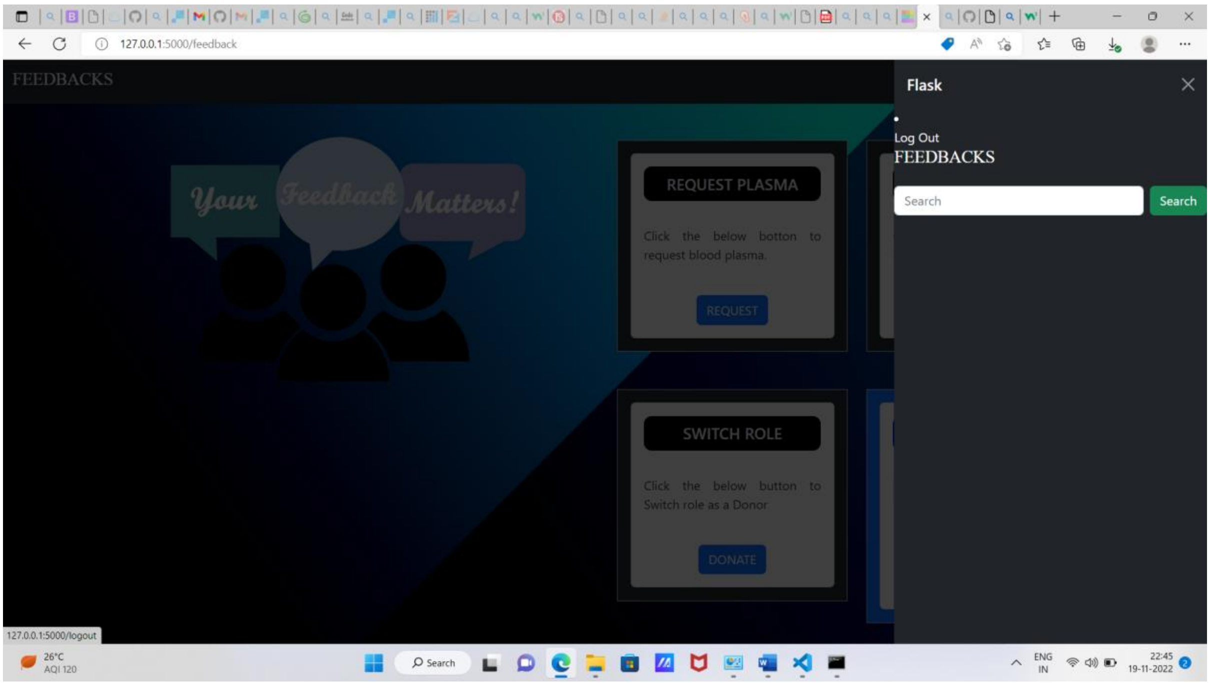
```
@app.route('/donorlist')
def donorlist():
    students = []
    sql = "SELECT * FROM donor"
    stmt = ibm_db.exec_immediate(conn, sql)
    dictionary = ibm_db.fetch_both(stmt)
    while dictionary != False:
        # print ("The Name is : ",  dictionary)
        students.append(dictionary)
        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('donordashboard.html',msg="Your details will be verified soon . Thankyou!")
if __name__ == '__main__':
    app.run(debug=True)
```

LOGOUT :

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

OUTPUT :

After login Each and Every page consists of logout option .

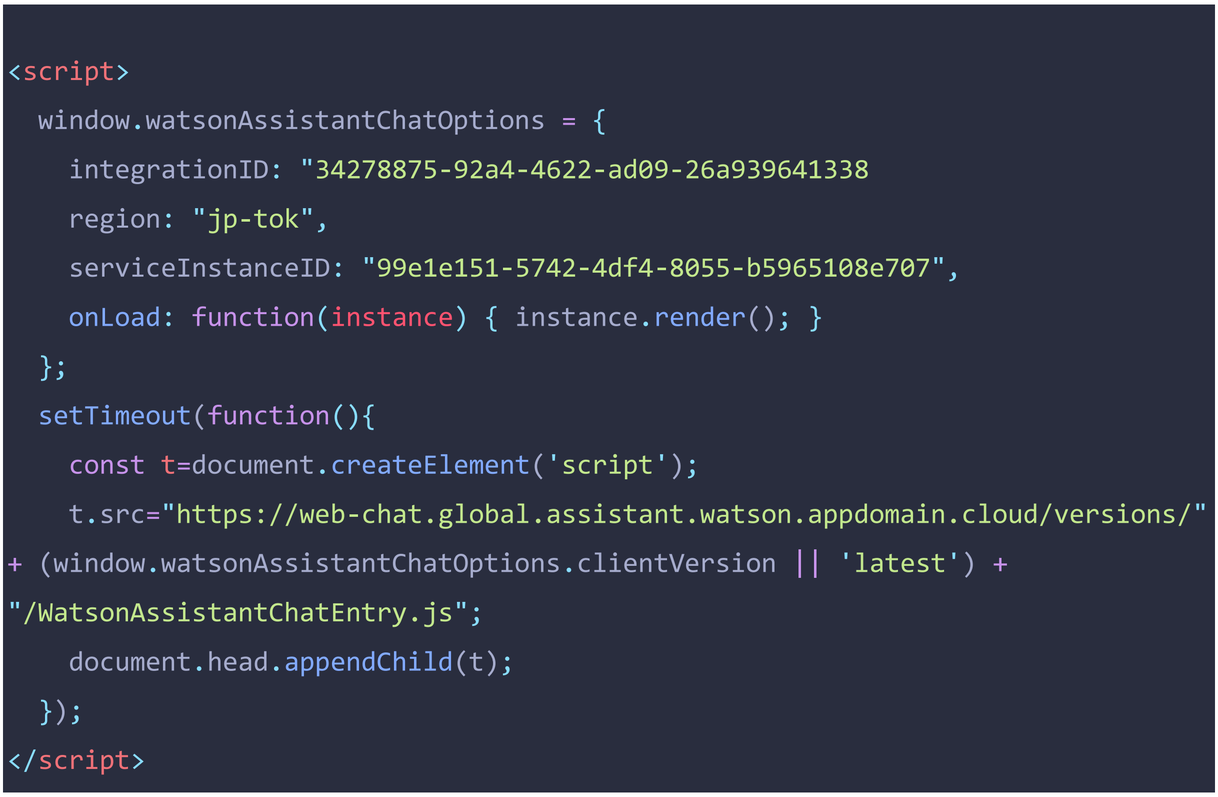


Click the below botton to request blood plasma.

Click the below button to Switch role as a Donor

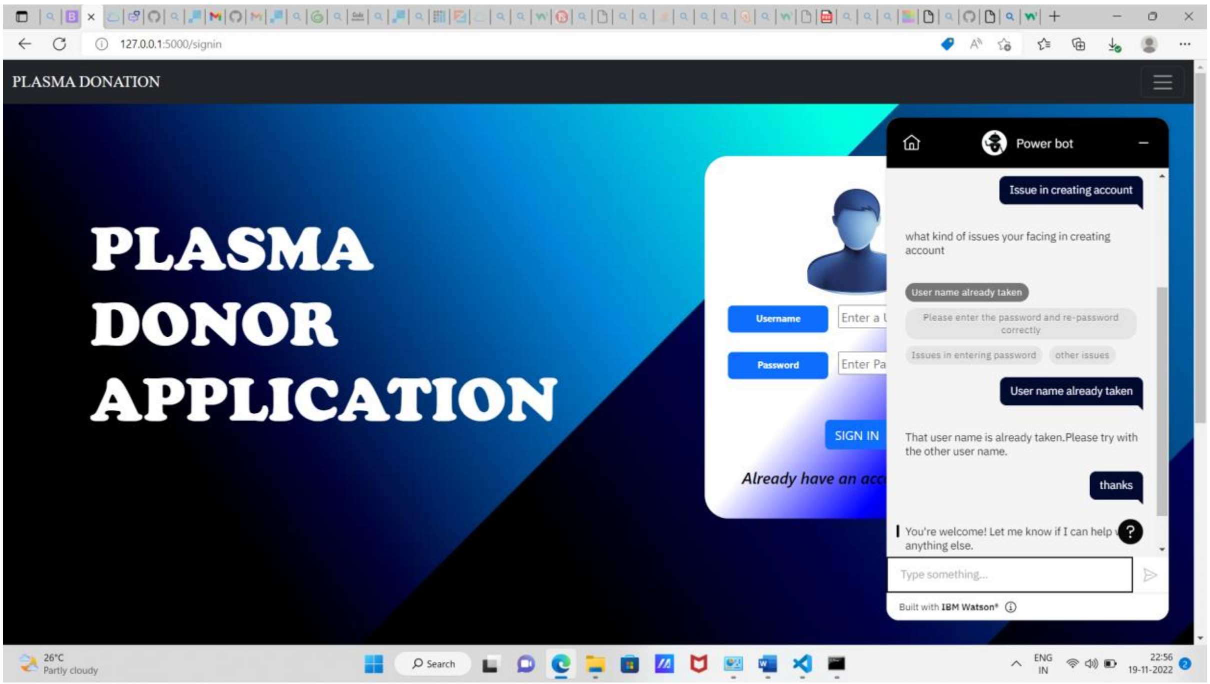
BOT SERVICE IN WEBSITE :

Embedded code of BOT in home.html :



```
<script>
  window.watsonAssistantChatOptions = {
    integrationID: "34278875-92a4-4622-ad09-26a939641338
    region: "jp-tok",
    serviceInstanceID: "99e1e151-5742-4df4-8055-b5965108e707",
    onLoad: function(instance) { instance.render(); }
  };
  setTimeout(function(){
    const t=document.createElement('script');
    t.src="https://web-chat.global.assistant.watson.appdomain.cloud/versions/"
+ (window.watsonAssistantChatOptions.clientVersion || 'latest') +
"/WatsonAssistantChatEntry.js";
    document.head.appendChild(t);
  });
</script>
```

OUTPUT:



Issue in creating account

what kind of issues your facing in creating account

User name already taken

Please enter the password and re-password correctly

Issues in entering password

other issues

User name already taken

That user name is already taken.Please try with the other user name.

thanks

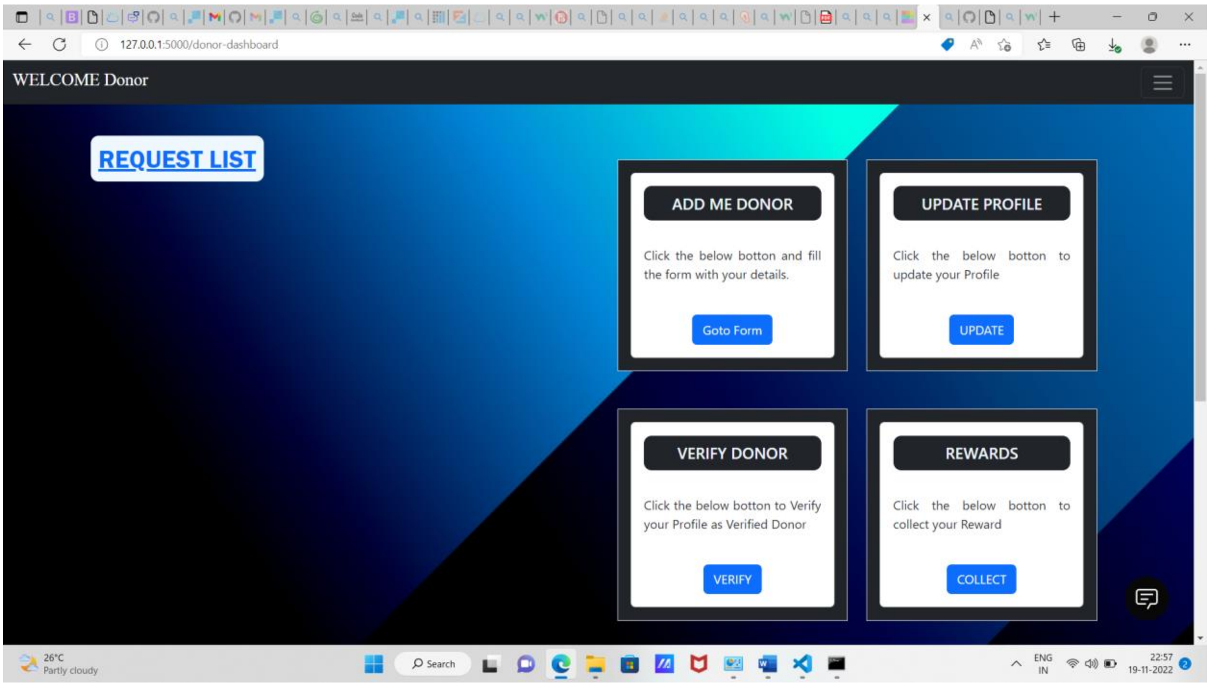
You're welcome! Let me know if I can help w anything else.

UPDATE PROFILE:

CODE:

```
{% extends "adddonor.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}}{{msg4}} </h6>
                {% endblock %}



 {% endblock %}
{% block form %}

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


{% endblock %}


{% block submit %}
<div class="btn-block">
    <button type="submit" value="submit" >Submit</button>
    </div>
{% endblock %}
```

OUTPUT:



127.0.0.1:5000/donor-dashboard

WELCOME Donor

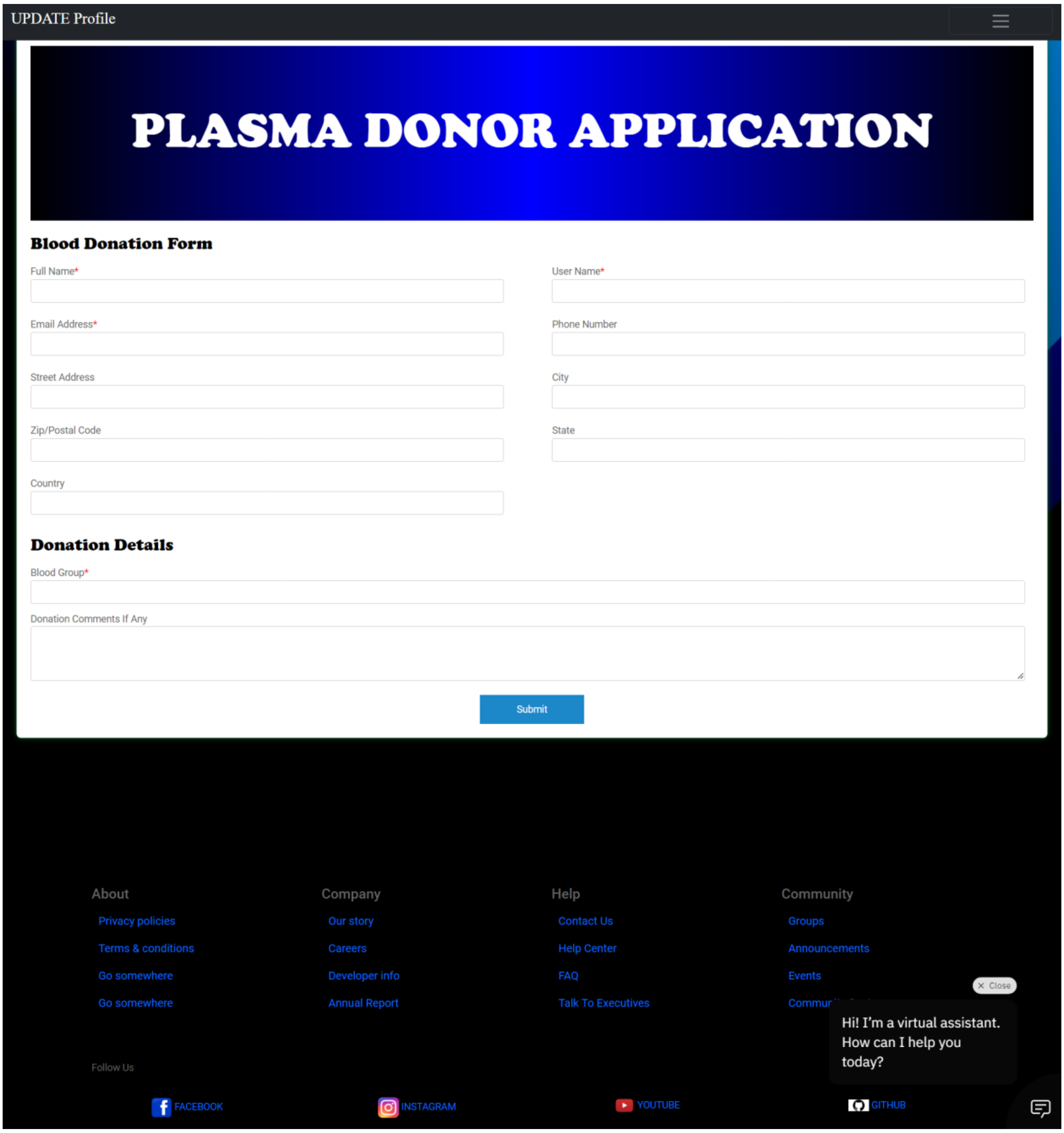
Click the below botton and fill the form with your details.

Click the below botton to update your Profile

Click the below botton to Verify your Profile as Verified Donor

Click the below botton to collect your Reward

User can Update their profile by clicking UPDATE PROFILE in donor dashboard.:



UPDATE Profile

PLASMA DONOR APPLICATION

Blood Donation Form

Full Name*

User Name*

Email Address*

Phone Number

Street Address

City

Zip/Postal Code

State

Country

Donation Details

Blood Group*

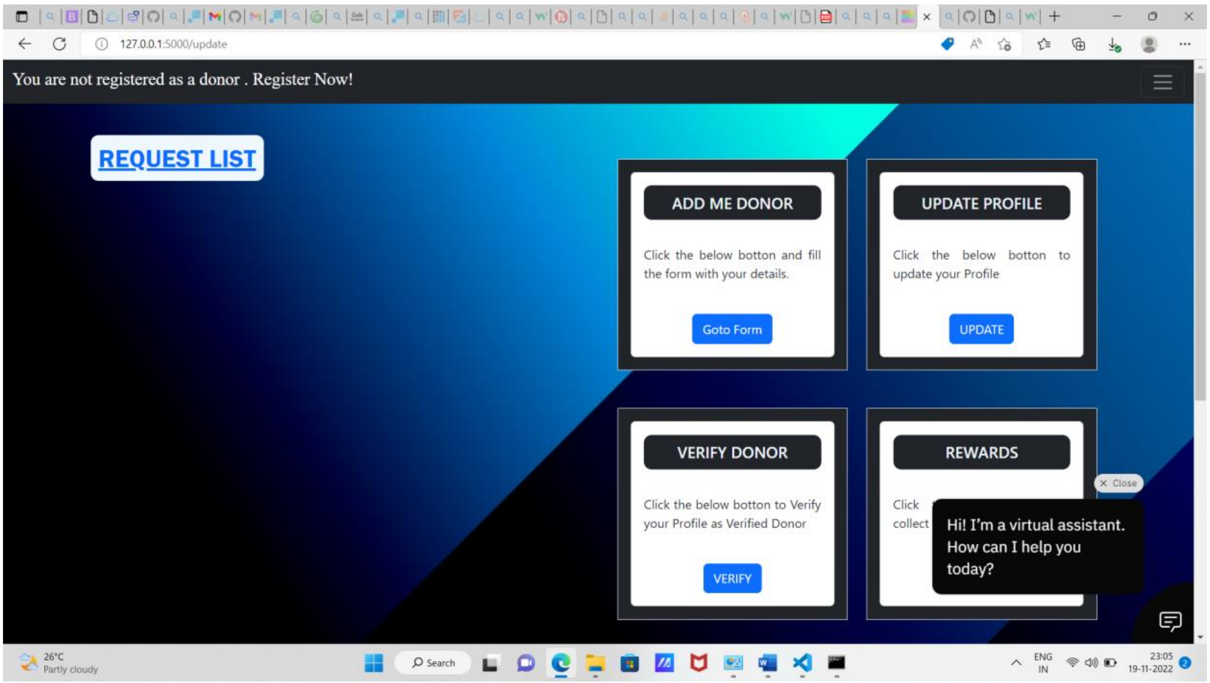
Donation Comments If Any

Submit

Hi! I'm a virtual assistant. How can I help you today?

TEST CASES:

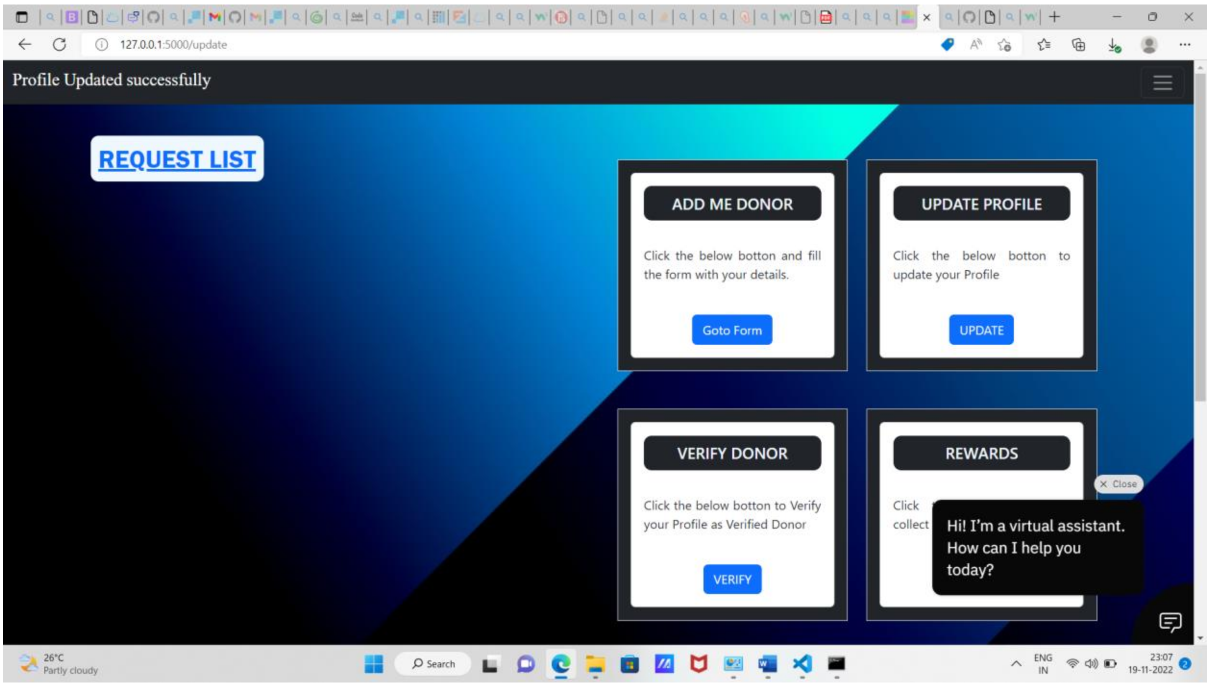
Not Yet Registered: If the donor not registered and try to update profile


127.0.0.1:5000/update

You are not registered as a donor . Register Now!

REQUEST LIST
ADD ME DONOR
Click the below botton and fill the form with your details.
Goto Form
UPDATE PROFILE
Click the below botton to update your Profile
UPDATE
VERIFY DONOR
Click the below botton to Verify your Profile as Verified Donor
VERIFY
REWARDS
Click
collect
× Close
Hi! I'm a virtual assistant. How can I help you today?
26°C Partly cloudy
Search
ENG IN
23:05 19-11-2022

Profile Updated Successfully:


127.0.0.1:5000/update

Profile Updated successfully

REQUEST LIST
ADD ME DONOR
Click the below botton and fill the form with your details.
Goto Form
UPDATE PROFILE
Click the below botton to update your Profile
UPDATE
VERIFY DONOR
Click the below botton to Verify your Profile as Verified Donor
VERIFY
REWARDS
Click
collect
× Close
Hi! I'm a virtual assistant. How can I help you today?
26°C Partly cloudy
Search
ENG IN
23:07 19-11-2022

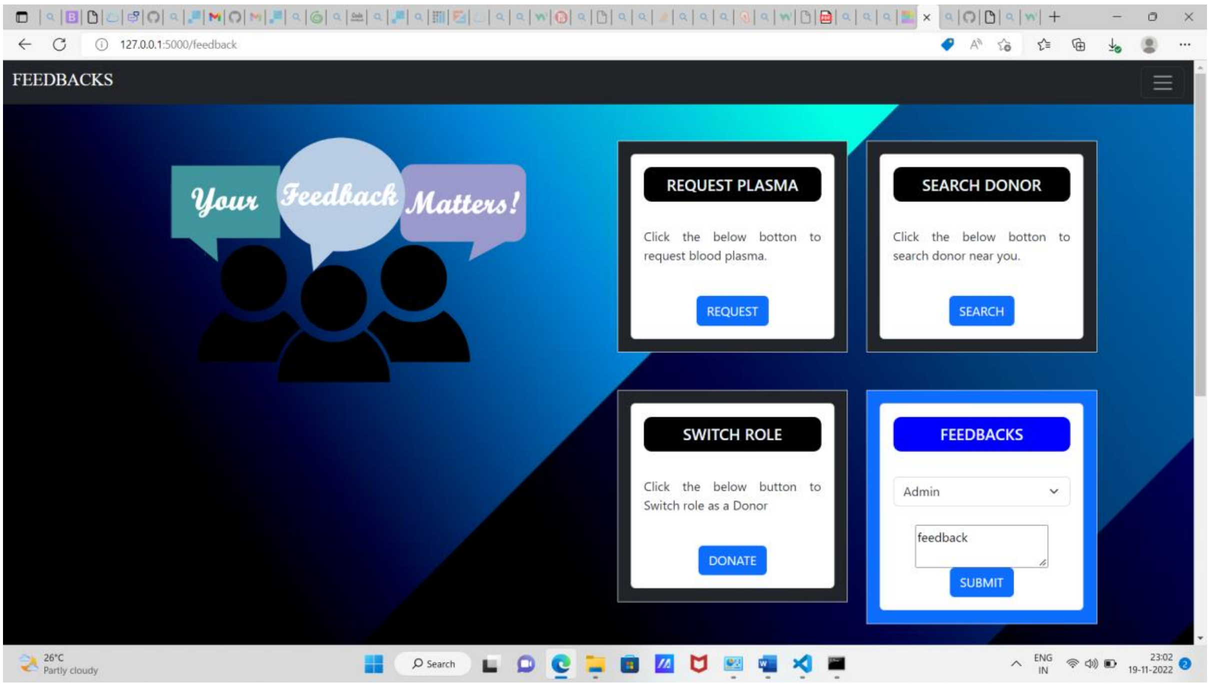
FEEDBACK:

CODE:

```
{% extends "receiverdashboard.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}}{{msg4}} </h6>
                {% endblock %}


 {% endblock %}
{% block donorlist %}
<BR>
    <img src="https://plasmasonation.s3.jp-tok.cloud-object-
storage.appdomain.cloud/feedback.png" width="500px;">
{% endblock %}
{% block feedback %}
<form action="feeding" 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>
```

```
<select class="form-select"  aria-label="size 3 select example" aria-placeholder="Select User" name="feedtouser">



  <option selected >Admin</option>
  {% for row in students %}
    <option  >{{row["USERNAME"]}}</option>
    {% endfor %}
  </select><br>

  <textarea name="feedback" placeholder="Type your feedback here">feedback
</textarea>

  <button type="submit" class="btn btn-primary" value="submit">SUBMIT</button>
</form>


{% endblock %}
```

OUTPUT:



127.0.0.1:5000/feedback

FEEDBACKS

REQUEST PLASMA

Click the below botton to request blood plasma.

SEARCH DONOR

Click the below botton to search donor near you.

SWITCH ROLE

Click the below botton to Switch role as a Donor

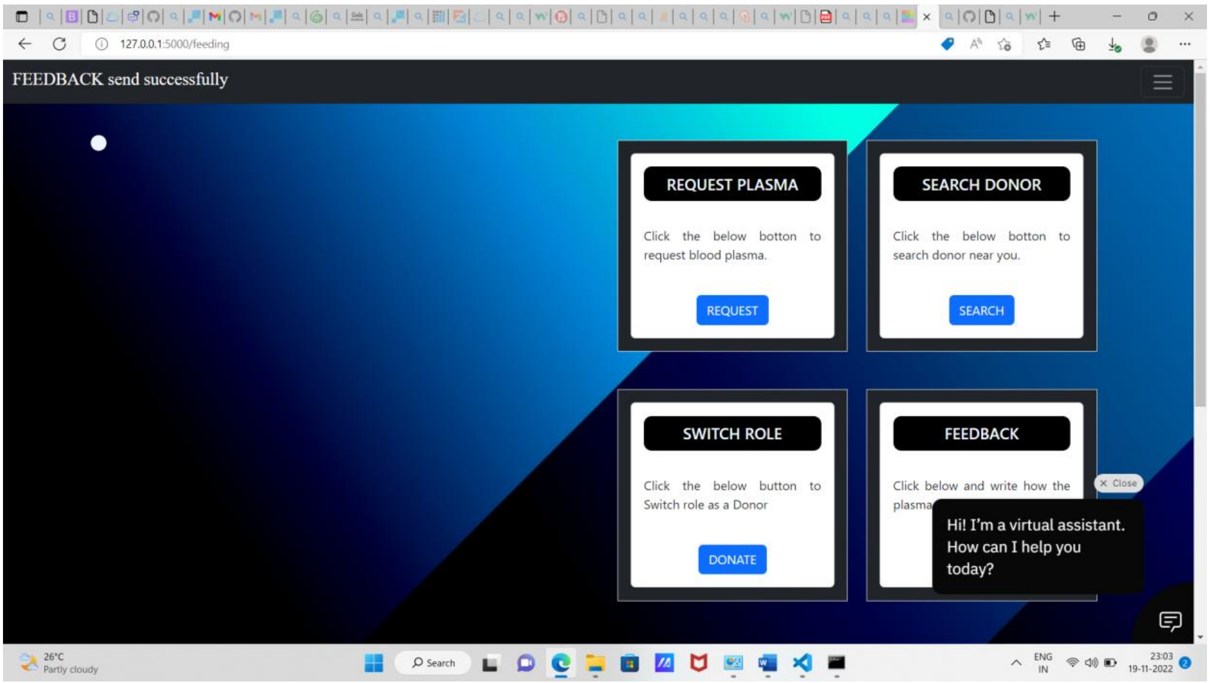
FEEDBACKS

Admin

feedback

TEST CASES:

FEEDBACK send successfully



FEEDBACK send successfully

REQUEST PLASMA

Click the below botton to request blood plasma.

SEARCH DONOR

Click the below botton to search donor near you.

SWITCH ROLE

Click the below button to Switch role as a Donor

FEEDBACK

Click below and write how the plasma

Hi! I'm a virtual assistant. How can I help you today?

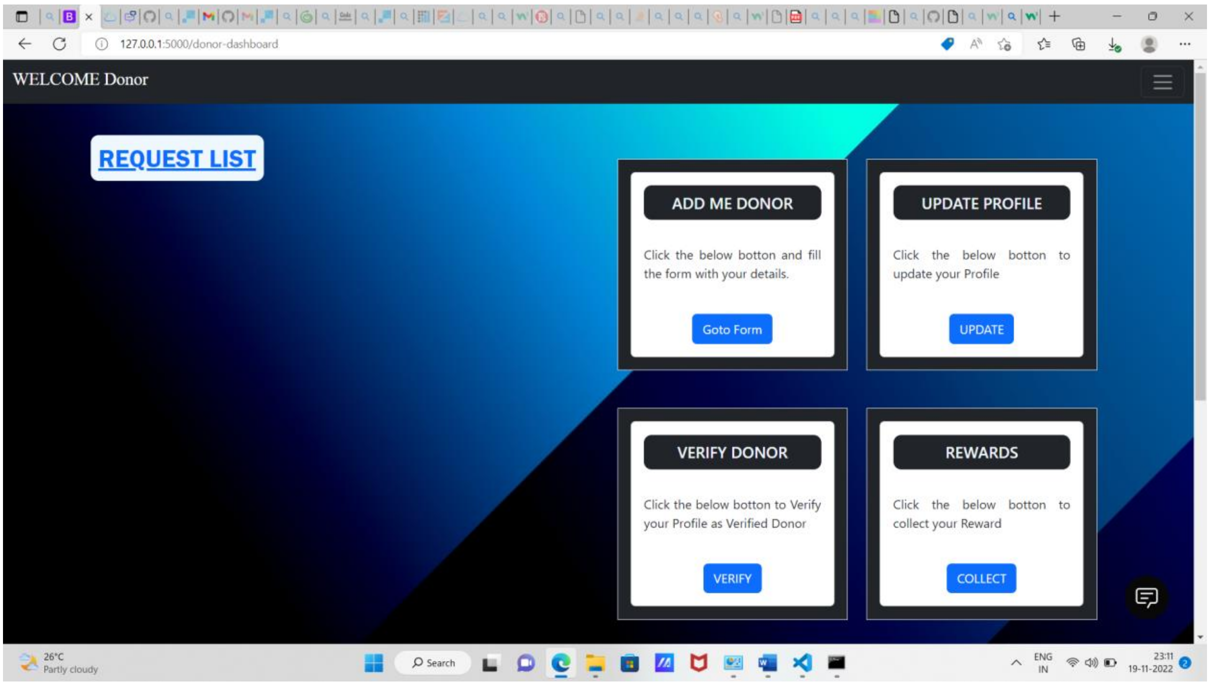
REWARDS :

CODE:

```
{% extends "donordashboard.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}}{{msg4}} </h6>
                {% endblock %}
 {% endblock %}

{% block reward%}
<form action="/certificate" method="post">
    <h5 class="card-title" style="background-color:blue;
color:aliceblue;border-radius :10px; padding:10px;">{{msg4}}</h5><br>

<input type="text" name="name" placeholder="Full name">
<input type="number" name="count" placeholder="No of Donation count"><br><br>
```

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

OUTPUT:



127.0.0.1:5000/donor-dashboard

WELCOME Donor

REQUEST LIST

ADD ME DONOR

Click the below botton and fill the form with your details.

Goto Form

UPDATE PROFILE

Click the below botton to update your Profile

UPDATE

VERIFY DONOR

Click the below botton to Verify your Profile as Verified Donor

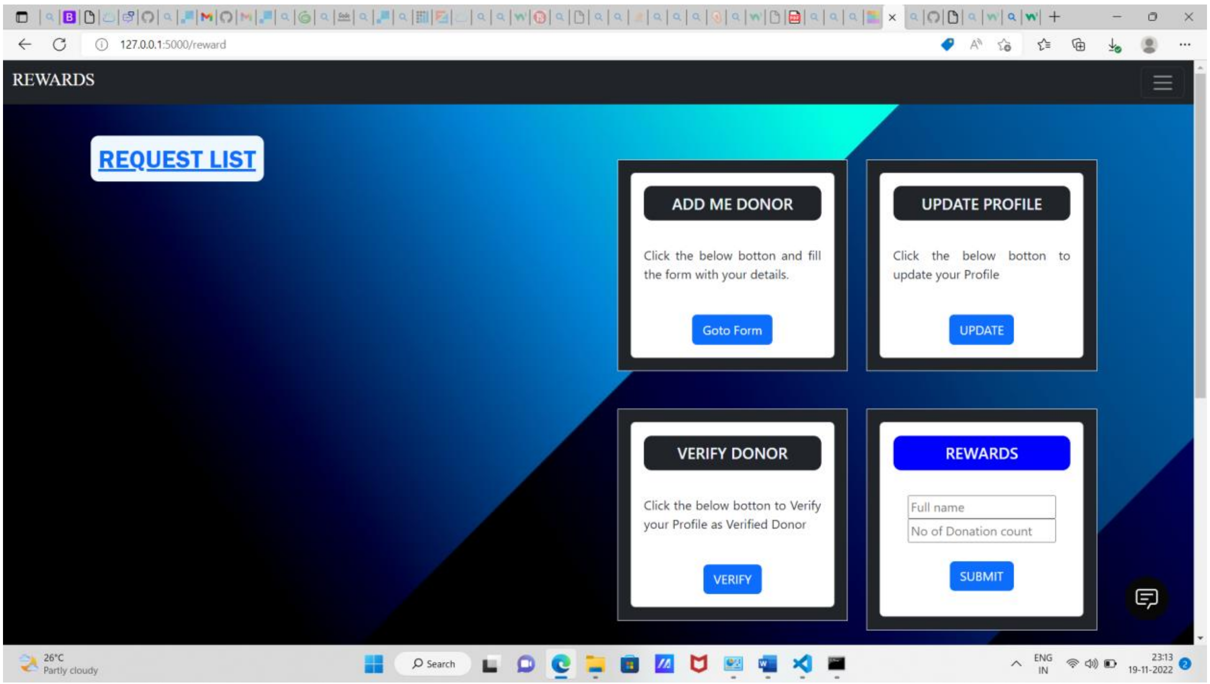
VERIFY

REWARDS

Click the below botton to collect your Reward

COLLECT

User can collect their Reward By Clicking Reward in Donor Dashboard .



127.0.0.1:5000/reward

REWARDS

REQUEST LIST

ADD ME DONOR

Click the below botton and fill the form with your details.

Goto Form

UPDATE PROFILE

Click the below botton to update your Profile

UPDATE

VERIFY DONOR

Click the below botton to Verify your Profile as Verified Donor

VERIFY

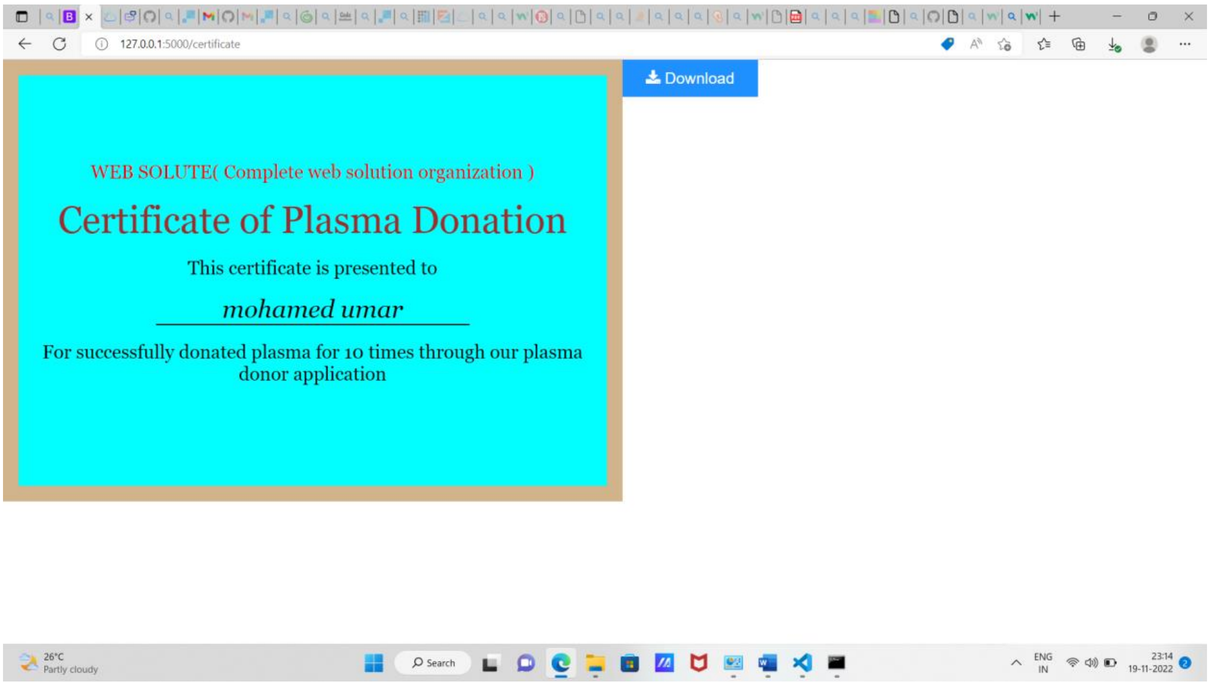
REWARDS

Full name

No of Donation count

SUBMIT

Submit Full Name and No.of Donation to claim certificate.



WEB SOLUTE(Complete web solution organization)

Certificate of Plasma Donation

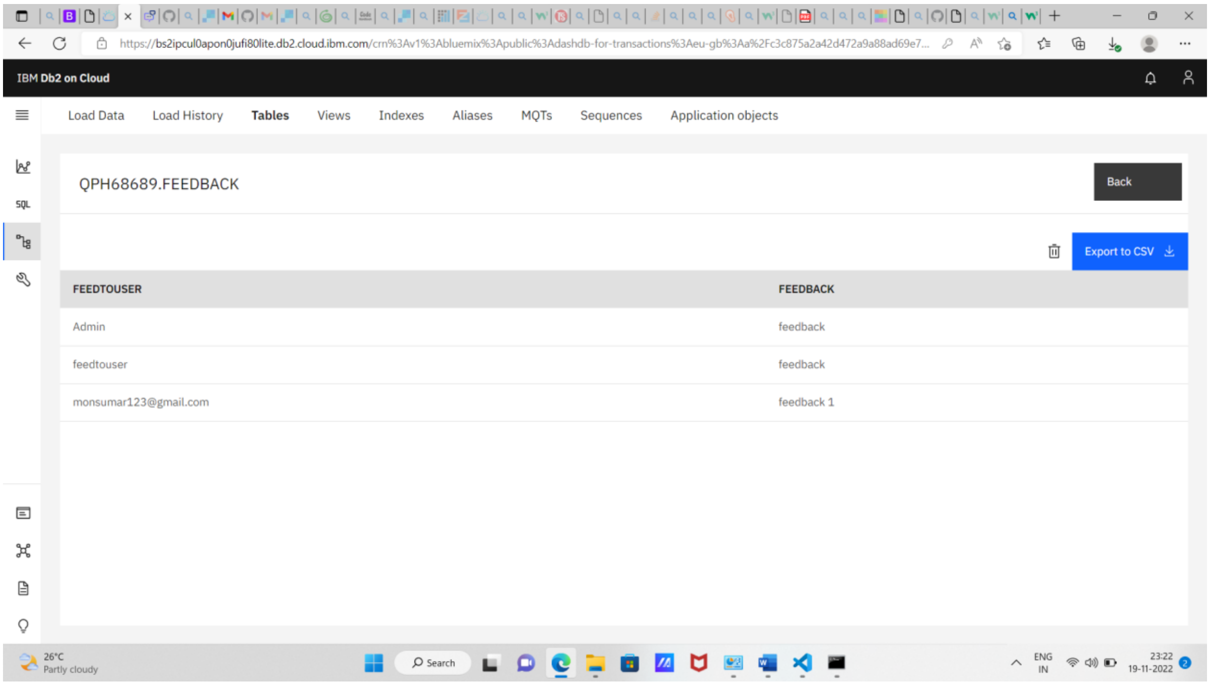
This certificate is presented to

mohamed umar

For successfully donated plasma for 10 times through our plasma donor application

DATABASE LIST :

FEEDBACKS:



QPH68689.FEEDBACK

FEEDTOUSER	FEEDBACK
Admin	feedback
feedtouser	feedback
monsumar123@gmail.com	feedback 1