```
from flask import Flask, redirect, url_for, render_template, request, make_response, jsonify, request
import ibm_db
from flask import request
import json

conn = ibm_db.connect(
   "DATABASE=bludb;HOSTNAME=764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=32536;SECURITY=SSL;SSLServerCertificate=abc.crt;UID=gnq12618;PWD=0glS4tFaR2ciK8fB",
   '', '')
print(conn)
print("connection successful...")
app = Flask(__name__)
import os
from sendgrid import SendGridAPIClient
from sendgrid.helpers.mail import Mail


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


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


@app.route('/login', methods=['POST', 'GET'])
def login():
   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)
role = str()
requests = []
if dic:
    role = dic['ROLE']
    # sql = "select * from user where blood_group=?"
    # stmt = ibm_db.prepare(conn, sql)
    # ibm_db.bind_param(stmt, 1, username)
    # ibm_db.execute(stmt)
    # dic = ibm_db.fetch_assoc(stmt)

    # while dic != False:
    #     single_request = {
    #         'name': dic['NAME'],
    #         'age': dic['AGE'],
    #         'sex': dic['SEX'],
    #         'blood_type': dic['BLOOD_TYPE']
    #     }
    #     print(single_request)
    #     requests.append(single_request)
    #     dic = ibm_db.fetch_assoc(stmt)
    return render_template('dashboard.html', username=username, role=role)
```

```
        else:
            return redirect(url_for('login'))
        return redirect(url_for('home'))
    elif request.method == 'GET':
        return render_template('login.html')


@app.route('/signup', methods=['POST', 'GET'])
def signup():
    if request.method == 'POST':
        username = request.form['username']
        email = request.form['email']
        password = request.form['password']
        roll_no = request.form['roll_no']
        sex = request.form['sex']
        age = request.form['age']
        address = request.form['address']
        blood_group = request.form['blood_group']
        sql = "insert into user 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, roll_no)
        ibm_db.bind_param(prep_stmt, 5, sex)
        ibm_db.bind_param(prep_stmt, 6, age)
        ibm_db.bind_param(prep_stmt, 7, "USER")
        ibm_db.bind_param(prep_stmt, 8, address)
        ibm_db.bind_param(prep_stmt, 9, blood_group)
        ibm_db.execute(prep_stmt)
        # db post operation
```

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


@app.route('/toggle', methods=['POST'])
def toggle_user():
    data = request.get_json(force=True)

    username = data['username']
    role = data['role']
    print(username)
    print(role)
    sql = "update user set role=? where username=?"
    prep_stmt = ibm_db.prepare(conn, sql)
    ibm_db.bind_param(prep_stmt, 1, role)
    ibm_db.bind_param(prep_stmt, 2, username)
    ibm_db.execute(prep_stmt)
    return jsonify(
        status="success",
        role=role
    )


@app.route('/requestPalsma', methods=['POST'])
def requestBloodPlasma():
    # fetch mail address of the donors
    data = request.get_json(force=True)
    username = data['username']
    name = data['name']
    age = data['age']
```

```
    sex = data['sex']
    blood_type = data['bloodtype']
    phone_number = data['phone_num']
    sql = "select email from user where blood_group=?"
    stmt = ibm_db.prepare(conn, sql)
    ibm_db.bind_param(stmt, 1, blood_type)
    ibm_db.execute(stmt)
    dic = ibm_db.fetch_assoc(stmt)
    email_list = []
    while dic != False:
        email_list.append(dic['EMAIL'])
        print(dic['EMAIL'])
        dic = ibm_db.fetch_assoc(stmt)
    # send mail


    message = Mail(
        from_email='eshwaran.s.2019.cse@rajalakshmi.edu.in',
        to_emails=email_list,
        subject='Sending with Twilio SendGrid is Fun',
        html_content='<h1>Need Of Blood</h1><table><tr><th>Name</th><th>' + name +
'</th></tr><tr><th>Age</th><th>' + age + '</th></tr><tr><th>Sex</th><th>' + sex +
'</th></tr><tr><th>Blood Group</th><th>' + blood_type + '</th></tr><tr><th>Phone
Number</th><th>' + phone_number + '</th></tr></table>'
    )
    try:
        sg = SendGridAPIClient("SG.3iBLSgAYTEuVbfSHu9dCPA.-
nrnikWJvaRlNLMONA04_CuKAyPeV69c46vPAh3vUX0")
        response = sg.send(message)
        print(response.status_code)
        print(response.body)
        print(response.headers)
    except Exception as e:
```

```
        print(e.message)
    # insert data into requests table
    sql = "insert into bloodrequests(username,name,age,sex,blood_type) values (?,?,?,?,?)"
    prep_stmt = ibm_db.prepare(conn, sql)
    ibm_db.bind_param(prep_stmt, 1, username)
    ibm_db.bind_param(prep_stmt, 2, name)
    ibm_db.bind_param(prep_stmt, 3, age)
    ibm_db.bind_param(prep_stmt, 4, sex)
    ibm_db.bind_param(prep_stmt, 5, blood_type)
    ibm_db.execute(prep_stmt)

    return jsonify(
        name=name,
        age=age,
        sex=sex,
        bloodtype=blood_type,
        status="yes"
    )


@app.route('/getrequests', methods=['POST'])
def getBloodRequests():
    data = request.get_json(force=True)
    username = data['username']
    sql = "select * from bloodrequests where username=?"
    stmt = ibm_db.prepare(conn, sql)
    ibm_db.bind_param(stmt, 1, username)
    ibm_db.execute(stmt)
    dic = ibm_db.fetch_assoc(stmt)
    requests = []
    print(type(dic))
```

```
    while dic != False:
        single_request = {
            'name': dic['NAME'],
            'age': dic['AGE'],
            'sex': dic['SEX'],
            'blood_type': dic['BLOOD_TYPE']
        }
        print(single_request)
        requests.append(single_request)
        dic = ibm_db.fetch_assoc(stmt)
    return jsonify(
        username=username,
        requests=requests
    )



if __name__ == '__main__':
    app.run(host="0.0.0.0", debug=True)
```