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=21fecfd8-47b7-4937-840d-d791d0218660.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=31864;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA;UID=sks13376;PWD=ihddRZZDMYcdcjin",'','')

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='keerthivasanece10@gmail.com',

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("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")

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)