from flask import Flask,render\_template,request,redirect,url\_for ,session

import ibm\_db

import re

import os

import math

import random

import smtplib

import requests

app=Flask(\_\_name\_\_,template\_folder='templates',static\_folder='static')

app.secret\_key='a'

conn = ibm\_db.connect("DATABASE=bludb;HOSTNAME=ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=31505;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=fbx49160;PWD=qiz4IhmLwIXrkXPv",'','')

print("successfully connected")

@app.route('/')

def home():

return render\_template('index.html')

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

def login():

global userid

msg=''

if request.method=='POST':

username=request.form.get('username',False)

password=request.form.get('password',False)

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)

account=ibm\_db.fetch\_assoc(stmt)

print(account)

if account:

session['Logged in']=True

session['id']=account['USERNAME']

userid=account['USERNAME']

session['username']=account['USERNAME']

msg='Logged in successfully'

return render\_template('dash.html')

else:

msg='Incorrect username/password'

return render\_template('login.html',msg=msg)

@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']

Firstname=request.form['firstname']

lastname=request.form['lastname']

#phoneno=request.form['phoneno']

sql='SELECT \* FROM USER WHERE username=?'

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:

msg="Account already exist!"

elif not re.match(r'[^@]+@[^@]+\.[^@]+',email):

msg="Invalid email address"

elif not re.match(r'[A-Za-z0-9]+',username):

msg="name must contain character and numbers"

else:

insert\_sql='INSERT INTO USER values(?,?,?,?,?)'

prep\_stmt=ibm\_db.prepare(conn, insert\_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,Firstname)

ibm\_db.bind\_param(prep\_stmt,5,lastname)

ibm\_db.execute(prep\_stmt)

msg="You have successfully registered"

return render\_template('verify.html',msg=msg)

elif request.method=="POST":

msg="Please fill out the form"

return render\_template('register.html',msg=msg)

@app.route('/welcome')

def welcome():

return render\_template('welcome.html')

@app.route('/verify')

def verify():

email=request.args.get('email', None)

server=smtplib.SMTP('smtp.gmail.com',587)

server.starttls()

password="nsgeuedwbzptosyp"

server.login(email,password)

otp=''.join([str(random.randint(0,9))for i in range(4)])

msg=' YOUR OTP IS'+str(otp)

server.sendmail(email,email,msg)

server.quit()

if request.method=='POST':

verify=request.method['code']

if verify==otp:

return render\_template('login.html')

return render\_template('verify.html')

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

def frgpwd():

msg =" "

print(request.form)

username1=request.form.get("uname", False)

oldpassword=request.form.get("oldpassword", False)

newpassword=request.form.get("newpassword", False)

sql='SELECT \* FROM USER WHERE username=?'

stmt=ibm\_db.prepare(conn,sql)

ibm\_db.bind\_param(stmt,1,username1)

#ibm\_db.bind\_param(stmt,2,password)

ibm\_db.execute(stmt)

account=ibm\_db.fetch\_assoc(stmt)

print(account)

if account:

chgpwd\_sql='UPDATE USER SET password = ? WHERE username = ?'

prep\_stmt=ibm\_db.prepare(conn, chgpwd\_sql)

ibm\_db.bind\_param(prep\_stmt,1,newpassword)

ibm\_db.bind\_param(prep\_stmt,2,username1)

ibm\_db.execute(prep\_stmt)

msg="You have successfully changed password"

return render\_template('forgot password.html',msg=msg)

return render\_template('forgot password.html',msg=msg)

url = "https://low-carb-recipes.p.rapidapi.com"

headers = {

"x-rapidapi-key": "ad933ea36amsh6b0a83e514b1a58p14bc9ejsne745a5851a1b",

"x-rapidapi-host": "low-carb-recipes.p.rapidapi.com"

}

searchForRecipes = "/search"

getRecipe="/recipes/"

getImage="/images/2807982c-986a-4def-9e3a-153a3066af7a.jpeg"

getRandomRecipe="/random"

@app.route('/login/dash')

def dashboard():

return render\_template('dash.html')

@app.route('/login/dash/viewprofile')

def viewprofile():

username=session['id']

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)

print(account)

if account:

return render\_template('viewprofile.html')

else:

return render\_template('personal info.html')

@app.route('/login/dash/viewprofile/personinfo',methods=['GET','POST'])

def per\_info():

msg=''

if request.method =='POST':

Name=request.form['Name']

gender=request.form['gender']

tar\_weight=request.form['Target Weight']

Age=request.form['Age']

Height=request.form['Height']

Weight=request.form['Weight']

email=request.form['email']

location=request.form['location']

phoneno=request.form['phoneno']

sql='SELECT \* FROM USER WHERE username=?'

stmt=ibm\_db.prepare(conn,sql)

ibm\_db.bind\_param(stmt,1,Name)

ibm\_db.execute(stmt)

account=ibm\_db.fetch\_assoc(stmt)

print(account)

if account:

insert\_sql='INSERT INTO USER values(?,?,?,?,?,?,?)'

prep\_stmt=ibm\_db.prepare(conn, insert\_sql)

ibm\_db.bind\_param(prep\_stmt,1,Name)

ibm\_db.bind\_param(prep\_stmt,2,gender)

ibm\_db.bind\_param(prep\_stmt,3,Age)

ibm\_db.bind\_param(prep\_stmt,4,Height)

ibm\_db.bind\_param(prep\_stmt,5,Weight)

ibm\_db.bind\_param(prep\_stmt,7,location)

ibm\_db.execute(prep\_stmt)

msg="Your details are successfully stored"

return render\_template('viewprofile.html',msg=msg)

elif request.method=="POST":

msg="Please fill out the form"

return render\_template('personal info.html',msg=msg)

@app.route('/login/dash/feedback',methods=['GET','POST'])

def feedback():

msg=''

if request.method =='POST':

Name=request.form['name']

email=request.form['email']

Feedback=request.form['Feedback']

sql='SELECT \* FROM USER WHERE username=?'

stmt=ibm\_db.prepare(conn,sql)

ibm\_db.bind\_param(stmt,1,Name)

ibm\_db.execute(stmt)

account=ibm\_db.fetch\_assoc(stmt)

print(account)

if account:

insert\_sql='INSERT INTO USER values(?,?,?)'

prep\_stmt=ibm\_db.prepare(conn, insert\_sql)

ibm\_db.bind\_param(prep\_stmt,1,Name)

ibm\_db.bind\_param(prep\_stmt,2,email)

ibm\_db.bind\_param(prep\_stmt,3,Feedback)

ibm\_db.execute(prep\_stmt)

msg="Your Feedback has been stored"

return render\_template('ratings.html',msg=msg)

elif request.method=="POST":

msg="Please fill out the form"

return render\_template('ratings.html',msg=msg)

@app.route('/dash/view recipe')

def search\_page():

#session ['item']=request.form.get("Ingridients", False)

return render\_template('search.html')

@app.route('/recipes')

def get\_recipes():

#food=session['item']

if (str(request.args['ingridients']).strip() != ""):

print(request.args['ingridients'])

# If there is a list of ingridients -> list

querystring = {"name":request.args['ingridients'],"tags":request.args['tag'],"includeIngredients":request.args['included'],"excludeIngredients":request.args['excluded'],"maxPrepareTime":"30","maxCookTime":"20","maxCalories":"500","maxNetCarbs":"5","maxSugar":"3","maxAddedSugar":"0","limit":"10"}

response = requests.request("GET", url + searchForRecipes, headers=headers, params=querystring)

data=response.json()

return render\_template('recipes.html', recipes=data)

else:

# Random recipes

response = requests.request("GET", url+ getRandomRecipe , headers=headers)

data=response.json()

return render\_template('recipes.html', recipes=data)

@app.route('/recipe')

def get\_recipe():

recipe\_id = request.args['id']

recipe\_info\_endpoint = "/recipes/{0}".format(recipe\_id)

print(recipe\_info\_endpoint)

recipe\_info = requests.request("GET", url + recipe\_info\_endpoint, headers=headers)

data=recipe\_info.json()

return render\_template('recipe.html', recipe=data)

@app.route('/logout')

def logout():

session.pop('loggedin',None)

session.pop('id',None)

session('username',None)

return render\_template("index.html")

if \_\_name\_\_=="\_\_main\_\_":

app.run(debug=True ,host='0.0.0.0',use\_reloader=False)