from flask import \*

import os

import ibm\_db

import bcrypt

from functools import partial,wraps

conn = ibm\_db.connect("DATABASE=;HOSTNAME=;PORT=32733;SECURITY=;SSLServerCertificate=;PROTOCOL=TCPIP;UID=;PWD=",'','')

app = Flask(\_\_name\_\_)

app.secret\_key = b'\_5#y2L"F4Q8z\n\xec]/'

PEOPLE\_FOLDER = os.path.join('static', 'people\_photo')

#IN THIS FILE ONLY SPRINT-2 ACTIONS ARE AVAILABLE

@app.route("/")

@app.route("/home")

#----------SPRINT-1-----------------#

@app.route('/logout')

def logout():

session.pop('loggedin', None)

session.pop('id', None)

return redirect(url\_for('home'))

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

#----------SPRINT-1-----------------#

@app.route("/user\_dashboard")

#----------SPRINT-1-----------------#

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

#----------SPRINT-1-----------------#

@app.route('/browse')

def addMarker():

if 'loggedin' in session:

query = "SELECT \* FROM jobpost;"

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

ibm\_db.execute(stmt)

a=[]

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

while(isUser!=False):

a.append(isUser)

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

else:

return redirect(url\_for('login'))

return render\_template("browse.html",result=a)

@app.route('/companies')

def companies():

if 'loggedin' in session:

query = "SELECT \* FROM RECRUITER"

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

ibm\_db.execute(stmt)

a=[]

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

while(isUser!=False):

a.append(isUser)

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

else:

return redirect(url\_for('login'))

return render\_template("companies.html",result=a)

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

def jobpost():

if 'loggedin' in session:

if request.method == 'POST':

recruiterid=request.form['recruiter\_id']

jobtitle = request.form['jobtitle']

jobtype = request.form['jobtype']

jobexp=request.form['jobexperience']

keyskill=request.form['keyskills']

location=request.form['location']

salary=request.form['salary']

discription=request.form['discription']

insert\_sql = "INSERT INTO JOBPOST (RECRUITER\_ID,JOBTITLE, JOBTYPE, EXPERIENCE, KEYSKILL, LOCATION, SALARY, DISCRIPTION) VALUES (?,?,?,?,?,?,?,?)"

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

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

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

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

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

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

ibm\_db.bind\_param(prep\_stmt, 6, location)

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

ibm\_db.bind\_param(prep\_stmt, 8, discription)

ibm\_db.execute(prep\_stmt)

else:

return redirect(url\_for('login'))

return render\_template("jobpost.html")

@app.route("/browse/searchjob",methods=['GET','POST'])

def searchjob():

if request.method=='POST':

searchopt=request.form['searchopt']

srctitle=request.form['srctitle']

query = "SELECT \* FROM JOBPOST WHERE "+searchopt+"="+chr(39)+srctitle+chr(39)

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

ibm\_db.execute(stmt)

a=[]

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

while(isUser!=False):

a.append(isUser)

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

return render\_template('browse.html',result=a)

if \_\_name\_\_ == "\_\_main\_\_": #checking if \_\_name\_\_'s value is '\_\_main\_\_'. \_\_name\_\_ is an python environment variable who's value will always be '\_\_main\_\_' till this is the first instatnce of app.py running

app.run(debug=True,port=8080,host= '192.168.43.233') #running flask (Initalised on line 4)