ASSIGNMENT 2

NAME: T.THIRUNILAINAYAGI

REG NO.: 211519104174

DOMAIN : CLOUD APPLICATION DEVELOPMENT

QUESTION: WORK ON SAMPLE JOB PORTAL APPLICATION CREATED

SOLUTION:

APP.PY

```
from flask import Flask, render_template, request, redirect, url_for, session

import ibm_db

import re

app = Flask(__name__)

app.secret_key = 'a'


#CONNECTION STATEMENT WITH IBM DB2 CLOUD
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=32733;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=bsb19147;PWD=M8Q6aCGQuLHwiHkU",",")

#ROUTING FUNCTION WITH RENDER TEMPLATE

@app.route('/')

def homer():

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

```
# ROUTING FUNCTION WITH GET AND POST METHODS FOR LOGIN-PAGE
@app.route('/login',methods =['GET', 'POST'])
def login():
    global userid
    msg = ‘’
        if request.method == 'POST' :
        username = request.form['username']
        password = request.form['password']
        sql = "SELECT * FROM users 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['loggedin'] = True
            session['id'] = account['USERNAME']
            userid=  account['USERNAME']
            session['username'] = account['USERNAME']
            msg = 'Logged in successfully !'
            return render_template('dashboard.html', msg = msg)
        else:
            msg = 'Incorrect username / password !'
    return render_template('login.html', msg = msg)
```

```
   #REGISTER PAGE
@app.route('/register', methods =['GET', 'POST'])
def registet():
    msg = ''
    if request.method == 'POST' :
        username = request.form['username']
        email = request.form['email']
        password = request.form['password']
        sql = "SELECT * FROM users 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:
            msg = 'Account already exists !'
        elif not re.match(r'[^@]+@[^@]+\.[^@]+', email):
            msg = 'Invalid email address !'
        elif not re.match(r'[A-Za-z0-9]+', username):
            msg = 'name must contain only characters and numbers !'
        else:
            insert_sql = "INSERT INTO  users 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.execute(prep_stmt)
            msg = 'You have successfully registered !'
    elif request.method == 'POST':
        msg = 'Please fill out the form !'
    return render_template('register.html', msg = msg)


@app.route('/dashboard')
def dash():


    return render_template('dashboard.html')

#JOB SEARCH
@app.route('/apply',methods =['GET', 'POST'])
def apply():
     msg = ''
     if request.method == 'POST' :
         username = request.form['username']
         email = request.form['email']


         qualification= request.form['qualification']
```

```
 skills = request.form['skills']
 jobs = request.form['s']
 sql = "SELECT * FROM users 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:
    msg = 'there is only 1 job position! for you'
    return render_template('apply.html', msg = msg)



#INSERTING DATA TO IBM DB2
 insert_sql = "INSERT INTO  job 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, qualification)
 ibm_db.bind_param(prep_stmt, 4, skills)
 ibm_db.bind_param(prep_stmt, 5, jobs)
 ibm_db.execute(prep_stmt)
 msg = 'You have successfully applied for job !'
 session['loggedin'] = True
```

```
         TEXT = "Hello Thirunilainayagi ,a new appliaction for job position"
+jobs+"is requested"


         #sendmail(TEXT,"thirunilai01@gmail.com")
         sendgridmail(" thirunilai01@gmail.com",TEXT)




     elif request.method == 'POST':
         msg = 'Please fill out the form !'
     return render_template('apply.html', msg = msg)


@app.route('/display')
def display():
    print(session["username"],session['id'])

    cursor = mysql.connection.cursor()
    cursor.execute('SELECT * FROM job WHERE userid = % s', (session['id'],))
    account = cursor.fetchone()
    print("accountdislay",account)



    return render_template('display.html',account = account)
```

```
@app.route('/logout')


def logout():
   session.pop('loggedin', None)
   session.pop('id', None)
   session.pop('username', None)
   return render_template('home.html')



    #default line similar to main method
if __name__ == '__main__':
   app.run(host='0.0.0.0')
```

The main purpose for which the assignment was given is to get practiced with Flask connection with IBM DB2. So I understood the logic behind it and the flask code. For assignment purpose I have included the app.py file by changing the portions which could be with my understanding.