ASSIGNMENT -2

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

import ibm\_db

import re

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

app.secret\_key = 'a'

conn = ibm\_db.connect("DATABASE=bludb;HOSTNAME=b70af05b-76e4-4bca-a1f5-23dbb4c6a74e.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=32716;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=jzc43091;PWD=PI8VtGRvZlSVT65A",'','')

@app.route('/')

def homer():

return render\_template('home.html')

@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 !'

msg = 'Logged in successfully !'

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

else:

msg = 'Incorrect username / password !'

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

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

defregistet():

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’[^@][+@[^@]+\.[^@](mailto:+@[%5e@]+\.%5b%5e@)]+’, 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 !’

elifrequest.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’)

@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)

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,a new application for job position” +jobs+”is requested”

Elifrequest.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’)

If \_\_name\_\_ == ‘\_\_main\_\_’:

App.run(host=’0.0.0.0’)