|  |  |
| --- | --- |
| **Team ID** | PNT2022TMID36073 |

**Main.py**

import random

import string

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

import ibm\_db

def Upper\_Lower\_string(length):

result = ''.join((random.choice(string.ascii\_uppercase)

for x in range(length)))

return result

conn = ibm\_db.connect("DATABASE=bludb;HOSTNAME=6667d8e9-9d4d-

4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;

PORT=30376;

SECURITY=SSL;

SSLServerCertificate=DigiCertGlobalRootCA.crt;

UID=dml49039;PWD=oviOR6wjSnNPcRC2", '', '')

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

app.config['SECRET\_KEY'] = 'helloworld'

@app.route('/')

def home():

return render\_template('index.html')

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

def register():

if request.method == "POST":

global rs

name = request.form.get('name')

email = request.form.get('email')

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

stmt = ibm\_db.prepare(conn, 'SELECT \* FROM user WHERE

username=?')

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

ibm\_db.execute(stmt)

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

print(rs)

if rs:

msg = 'Account already Exists'

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

else:

reg\_stmt = ibm\_db.prepare(

conn, 'INSERT INTO user

("USERNAME","EMAIL","PASSWORD") VALUES(?,?,?)')

ibm\_db.bind\_param(reg\_stmt, 1, name)

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

ibm\_db.bind\_param(reg\_stmt, 3, password)

ibm\_db.execute(reg\_stmt)

msg = 'Successfully Registered'

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

else:

return render\_template('register.html')

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

def login():

if request.method == "POST":

customer = list()

agent = list()

name = request.form['name']

password = request.form['password']

log\_stmt = ibm\_db.prepare(

conn, 'SELECT \* FROM user WHERE username=? and password=?')

ibm\_db.bind\_param(log\_stmt, 1, name)

ibm\_db.bind\_param(log\_stmt, 2, password)

ibm\_db.execute(log\_stmt)

rs = ibm\_db.fetch\_assoc(log\_stmt)

if rs:

session['role'] = 'user'

session['customer'] = rs

print(rs)

return render\_template('dashboard.html')

log\_stmt = ibm\_db.prepare(

conn, 'SELECT \* FROM agent WHERE username=? and password=?')

ibm\_db.bind\_param(log\_stmt, 1, name)

ibm\_db.bind\_param(log\_stmt, 2, password)

ibm\_db.execute(log\_stmt)

rs = ibm\_db.fetch\_assoc(log\_stmt)

if rs:

cms = ibm\_db.exec\_immediate(conn, 'SELECT \* FROM user')

agt = ibm\_db.exec\_immediate(conn, 'SELECT \* FROM agent')

customers = ibm\_db.fetch\_assoc(cms)

agents = ibm\_db.fetch\_assoc(agt)

while customers:

customer.append(customers)

customers = ibm\_db.fetch\_assoc(cms)

while agents:

agent.append(agents)

agents = ibm\_db.fetch\_assoc(agt)

print(customer)

print(agent)

session['role'] = 'agent'

session['name'] = rs['USERNAME']

session['customer'] = customer

session['agent'] = agent

return render\_template('dashboard.html')

log\_stmt = ibm\_db.prepare(

conn, 'SELECT \* FROM admin WHERE username=? and password=?')

ibm\_db.bind\_param(log\_stmt, 1, name)

ibm\_db.bind\_param(log\_stmt, 2, password)

ibm\_db.execute(log\_stmt)

rs = ibm\_db.fetch\_assoc(log\_stmt)

if rs:

cms = ibm\_db.exec\_immediate(conn, 'SELECT \* FROM user')

agt = ibm\_db.exec\_immediate(conn, 'SELECT \* FROM agent')

customers = ibm\_db.fetch\_assoc(cms)

agents = ibm\_db.fetch\_assoc(agt)

while customers:

customer.append(customers)

customers = ibm\_db.fetch\_assoc(cms)

while agents:

agent.append(agents)

agents = ibm\_db.fetch\_assoc(agt)

print(customer)

print(agent)

session['role'] = 'admin'

session['customer'] = customer

session['agent'] = agent

return render\_template('dashboard.html', agent=agent,

customer=customer)

else:

msg = 'UID/Password is incorrect'

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

else:

return render\_template('login.html')

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

def dashboard():

return render\_template('dashboard.html')

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

def success():

if request.method == "POST":

ticket = session['ticket'] = Upper\_Lower\_string(16)

print(ticket, session['ticket'])

query = request.form['query']

sql = "UPDATE user SET QUERY=?,TICKET=? WHERE

USERNAME=?"

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

ibm\_db.bind\_param(out, 1, query)

ibm\_db.bind\_param(out, 2, session['ticket'])

ibm\_db.bind\_param(out, 3, session['name'])

status = ibm\_db.execute(out)

if status:

msg = 'Success ! Your Ticket Nno is :', ticket, 'You can now return to

the home page'

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

else:

msg = 'Error Submitting your Query'

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

@app.route('/redirect')

def redir():

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

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

def admin\_query():

msg = ""

agent = request.form.getlist('agent\_name')

usr\_name = request.form.getlist('cus\_name')

for i in range(0, len(agent)):

if agent[i] != 'none':

qr = ibm\_db.prepare(

conn, "UPDATE USER SET ASSIGNED\_AGENT=? WHERE

USERNAME=?")

ibm\_db.bind\_param(qr, 1, agent[i])

ibm\_db.bind\_param(qr, 2, usr\_name[i])

result = ibm\_db.execute(qr)

print(agent[i], " ", usr\_name[i])

if result:

msg = '<h1>queries executed</h1>'

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

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

def agent\_submit\_reply():

names = request.form.getlist('name')

text = request.form.getlist('text')

print(names)

print(text)

for i in range(0, len(names)):

if not text[i] == '':

try:

sql = 'UPDATE USER SET REPLY=?,REVIEW\_STATUS=1

WHERE USERNAME=?'

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

ibm\_db.bind\_param(query, 1, text[i])

ibm\_db.bind\_param(query, 2, names[i])

ibm\_db.execute(query)

except:

print('an error occured')

return '<html><body>done</body></html>'

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=True)