**Docker file**

FROM python:3.10.4

WORKDIR /app

COPY requirements.txt ./

RUN pip install -r requirements.txt

COPY . .

EXPOSE 5000

CMD ["python","./app.py"]

**app.py**

from \_\_future\_\_ import print\_function

from audioop import add

import datetime

from unicodedata import name

from sib\_api\_v3\_sdk.rest import ApiException

from pprint import pprint

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

from markupsafe import escape

from flask import \*

import ibm\_db

import datetime

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

print(conn)

print("connection successful...")

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

app.secret\_key = 'your secret key'

@app.route('/')

def home():

message = "TEAM ID : PNT2022TMID05042" +" "+ "BATCH ID : B3-3M5E "

return render\_template('index.html',mes=message)

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

def index():

return render\_template('index.html')

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

def signinpage():

return render\_template('signinpage.html')

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

def agentsignin():

return render\_template('signinpageagent.html')

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

def signuppage():

return render\_template('signuppage.html')

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

def agentRegister():

return render\_template('agentregister.html')

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

def forgotpass():

return render\_template('forgot.html')

@app.route('/newissue/<name>', methods=['POST', 'GET'])

def newissue(name):

name = name

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

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

def forgot():

try:

global randomnumber

ida = request.form['custid']

print(ida)

global id

id = ida

sql = "SELECT EMAIL,NAME FROM Customer WHERE id=?"

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

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

ibm\_db.execute(stmt)

emailf = ibm\_db.fetch\_both(stmt)

while emailf != False:

e = emailf[0]

n = emailf[1]

break

configuration = sib\_api\_v3\_sdk.Configuration()

configuration.api\_key['api-key'] = "null"

api\_instance = sib\_api\_v3\_sdk.TransactionalEmailsApi(

sib\_api\_v3\_sdk.ApiClient(configuration))

subject = "Verification for Password"

html\_content = "<html><body><h1>Your verification Code is : <h2>" + \

str(randomnumber)+"</h2> </h1> </body></html>"

sender = {"name": "IBM CUSTOMER CARE REGISTRY",

"email": "ibmdemo6@yahoo.com"}

to = [{"email": e, "name": n}]

reply\_to = {"email": "ibmdemo6@yahoo.com", "name": "IBM"}

headers = {"Some-Custom-Name": "unique-id-1234"}

params = {"parameter": "My param value",

"subject": "Email Verification"}

send\_smtp\_email = sib\_api\_v3\_sdk.SendSmtpEmail(

to=to, reply\_to=reply\_to, headers=headers, html\_content=html\_content, params=params, sender=sender, subject=subject)

api\_response = api\_instance.send\_transac\_email(send\_smtp\_email)

pprint(api\_response)

message = "Email send to:"+e+" for password"

flash(message, "success")

except ApiException as e:

print("Exception when calling SMTPApi->send\_transac\_email: %s\n" % e)

flash("Error in sending mail")

except:

flash("Your didn't Signin with this account")

finally:

return render\_template('forgot.html')

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

def agentforgot():

try:

global randomnumber

ida = request.form['custid']

print(ida)

global id

id = ida

sql = "SELECT EMAIL,NAME FROM AGENT WHERE id=?"

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

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

ibm\_db.execute(stmt)

emailf = ibm\_db.fetch\_both(stmt)

while emailf != False:

e = emailf[0]

n = emailf[1]

break

configuration = sib\_api\_v3\_sdk.Configuration()

configuration.api\_key['api-key'] = "null"

api\_instance = sib\_api\_v3\_sdk.TransactionalEmailsApi(

sib\_api\_v3\_sdk.ApiClient(configuration))

subject = "Verification for Password"

html\_content = "<html><body><h1>Your verification Code is : <h2>" + \

str(randomnumber)+"</h2> </h1> </body></html>"

sender = {"name": "IBM CUSTOMER CARE REGISTRY",

"email": "ibmdemo6@yahoo.com"}

to = [{"email": e, "name": n}]

reply\_to = {"email": "ibmdemo6@yahoo.com", "name": "IBM"}

headers = {"Some-Custom-Name": "unique-id-1234"}

params = {"parameter": "My param value",

"subject": "Email Verification"}

send\_smtp\_email = sib\_api\_v3\_sdk.SendSmtpEmail(

to=to, reply\_to=reply\_to, headers=headers, html\_content=html\_content, params=params, sender=sender, subject=subject)

api\_response = api\_instance.send\_transac\_email(send\_smtp\_email)

pprint(api\_response)

message = "Email send to:"+e+" for OTP"

flash(message, "success")

except ApiException as e:

print("Exception when calling SMTPApi->send\_transac\_email: %s\n" % e)

flash("Error in sending mail")

except:

flash("Your didn't Signin with this account")

finally:

return render\_template('forgot.html')

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

def admin():

userdatabase = []

sql = "SELECT \* FROM customer"

stmt = ibm\_db.exec\_immediate(conn, sql)

dictionary = ibm\_db.fetch\_both(stmt)

while dictionary != False:

userdatabase.append(dictionary)

dictionary = ibm\_db.fetch\_both(stmt)

if userdatabase:

sql = "SELECT COUNT(\*) FROM customer;"

stmt = ibm\_db.exec\_immediate(conn, sql)

user = ibm\_db.fetch\_both(stmt)

users = []

sql = "select \* from ISSUE"

stmt = ibm\_db.exec\_immediate(conn, sql)

dict = ibm\_db.fetch\_both(stmt)

while dict != False:

users.append(dict)

dict = ibm\_db.fetch\_both(stmt)

if users:

sql = "SELECT COUNT(\*) FROM ISSUE;"

stmt = ibm\_db.exec\_immediate(conn, sql)

count = ibm\_db.fetch\_both(stmt)

agent = []

sql = "SELECT \* FROM AGENT"

stmt = ibm\_db.exec\_immediate(conn, sql)

dictionary = ibm\_db.fetch\_both(stmt)

while dictionary != False:

agent.append(dictionary)

dictionary = ibm\_db.fetch\_both(stmt)

if agent:

sql = "SELECT COUNT(\*) FROM AGENT;"

stmt = ibm\_db.exec\_immediate(conn, sql)

cot = ibm\_db.fetch\_both(stmt)

return render\_template("admin.html",complaint=users,users=userdatabase,agents=agent,message=user[0],issue=count[0],msgagent = cot[0])

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

def remove():

otp = request.form['otpv']

if otp == 'C':

try:

insert\_sql = f"delete from customer"

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

ibm\_db.execute(prep\_stmt)

flash("delected successfully the Customer", "success")

except:

flash("No data found in Customer", "danger")

finally:

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

if otp == 'A':

try:

insert\_sql = f"delete from AGENT"

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

ibm\_db.execute(prep\_stmt)

flash("delected successfully the Agents", "success")

except:

flash("No data found in Agents", "danger")

finally:

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

if otp == 'C':

try:

insert\_sql = f"delete from AGENT"

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

ibm\_db.execute(prep\_stmt)

flash("delected successfully the Complaints", "success")

except:

flash("No data found in Complaints", "danger")

finally:

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

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

def login():

if request.method == 'POST':

try:

id = request.form['idn']

global hello

hello = id

password = request.form['password']

print(id, password)

if id == '1111' and password == '1111':

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

sql = f"select \* from customer where id='{escape(id)}' and password='{escape(password)}'"

stmt = ibm\_db.exec\_immediate(conn, sql)

data = ibm\_db.fetch\_both(stmt)

if data:

session["name"] = escape(id)

session["password"] = escape(password)

return redirect(url\_for("welcome"))

else:

flash("Mismatch in credetials", "danger")

except:

flash("Error in Insertion operation", "danger")

return render\_template('signinpage.html')

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

def welcome():

try:

id = hello

sql = "SELECT ID,DATE,TOPIC,SERVICE\_TYPE,SERVICE\_AGENT,DESCRIPTION,STATUS FROM ISSUE WHERE CUSTOMER\_ID =?"

agent = []

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

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

ibm\_db.execute(stmt)

otpf = ibm\_db.fetch\_both(stmt)

while otpf != False:

agent.append(otpf)

otpf = ibm\_db.fetch\_both(stmt)

sql = "SELECT COUNT(\*) FROM ISSUE WHERE CUSTOMER\_ID = ?"

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

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

ibm\_db.execute(stmt)

t = ibm\_db.fetch\_both(stmt)

return render\_template("welcome.html",agent=agent,message=t[0])

except:

return render\_template("welcome.html")

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

def loginagent():

if request.method == 'POST':

try:

global loginagent

id = request.form['idn']

loginagent = id

password = request.form['password']

sql = f"select \* from AGENT where id='{escape(id)}' and password='{escape(password)}'"

stmt = ibm\_db.exec\_immediate(conn, sql)

data = ibm\_db.fetch\_both(stmt)

if data:

session["name"] = escape(id)

session["password"] = escape(password)

return redirect(url\_for("agentwelcome"))

else:

flash("Mismatch in credetials", "danger")

except:

flash("Error in Insertion operation", "danger")

return render\_template("signinpageagent.html")

@app.route('/delete/<ID>')

def delete(ID):

sql = f"select \* from customer where Id='{escape(ID)}'"

print(sql)

stmt = ibm\_db.exec\_immediate(conn, sql)

student = ibm\_db.fetch\_row(stmt)

if student:

sql = f"delete from customer where id='{escape(ID)}'"

stmt = ibm\_db.exec\_immediate(conn, sql)

flash("Delected Successfully", "success")

return redirect(url\_for("admin"))

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

def agentform():

if request.method == 'POST':

try:

x = datetime.datetime.now()

y = x.strftime("%Y-%m-%d %H:%M:%S")

name1 = request.form['name']

email = request.form['email']

password = request.form['password']

phonenumber = request.form['phonenumber']

service = request.form['service']

address = request.form['address']

city = request.form['city']

state = request.form['state']

country = request.form['country']

link = request.form['link']

sql = "SELECT \* FROM AGENT WHERE EMAIL = ?"

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

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

ibm\_db.execute(stmt)

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

if account:

flash("Record Aldready found", "success")

else:

print("exec")

insert\_sql = "INSERT INTO AGENT (NAME,EMAIL,PASSWORD,PHONENUMBER,SERVICE\_AGENT,ADDRESS,CITY,STATE,COUNTRY,RESUME\_LINK,DATE) VALUES(?,?,?,?,?,?,?,?,?,?,?)"

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

ibm\_db.bind\_param(prep\_stmt, 1, name1)

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

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

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

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

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

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

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

ibm\_db.bind\_param(prep\_stmt, 9, country)

ibm\_db.bind\_param(prep\_stmt, 10, link)

ibm\_db.bind\_param(prep\_stmt, 11, y)

ibm\_db.execute(prep\_stmt)

flash("Record stored Successfully", "success")

sql = "SELECT ID FROM AGENT WHERE email=?"

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

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

ibm\_db.execute(stmt)

hi = ibm\_db.fetch\_tuple(stmt)

configuration = sib\_api\_v3\_sdk.Configuration()

configuration.api\_key['api-key'] = "null"

api\_instance = sib\_api\_v3\_sdk.TransactionalEmailsApi(

sib\_api\_v3\_sdk.ApiClient(configuration))

subject = "Registering Account in Customer Care Registry"

html\_content = " <html><body><h1>Thanks for Registering into Customer Care Registry</h1> <h2>Your Account Id is :"+str(hi[0])+"</h2><h2>With Regards:</h2><h3>Customer Care Registry</h3> </body></html>"

sender = {"name": "IBM CUSTOMER CARE REGISTRY",

"email": "ibmdemo6@yahoo.com"}

to = [{"email": email, "name": name1}]

reply\_to = {"email": "ibmdemo6@yahoo.com", "name": "IBM"}

headers = {"Some-Custom-Name": "unique-id-1234"}

params = {"parameter": "My param value",

"subject": "Email Verification"}

send\_smtp\_email = sib\_api\_v3\_sdk.SendSmtpEmail(

to=to, reply\_to=reply\_to, headers=headers, html\_content=html\_content, params=params, sender=sender, subject=subject)

api\_response = api\_instance.send\_transac\_email(send\_smtp\_email)

pprint(api\_response)

except:

flash("Error in Insertion Operation", "danger")

finally:

return redirect(url\_for("agentRegister"))

con.close()

return render\_template('agentregister.html')

@app.route('/completed/<DESCRIPTION>', methods=['GET', 'POST'])

def completed(DESCRIPTION):

status ="Completed"

try:

sql = "UPDATE ISSUE SET STATUS = ? WHERE DESCRIPTION =?"

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

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

ibm\_db.bind\_param(stmt,2,DESCRIPTION)

ibm\_db.execute(stmt)

flash("Successful","success")

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

except:

flash("No record found","danger")

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

@app.route('/deletecomplaint/<ID>')

def deletecomplaint(ID):

sql = f"select \* from ISSUE where ID='{escape(ID)}'"

print(sql)

stmt = ibm\_db.exec\_immediate(conn, sql)

student = ibm\_db.fetch\_row(stmt)

if student:

sql = f"delete from ISSUE where ID='{escape(ID)}'"

stmt = ibm\_db.exec\_immediate(conn, sql)

users = []

flash("Delected Successfully", "success")

return redirect(url\_for("admin"))

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

app.run(host='0.0.0.0', port=5000, debug=True)

**int.py**

import random

randomnumber = random.randint(0,100)

id =0

hello =0

**Requirements**

flask

ibm\_db

sib\_api\_v3\_sdk