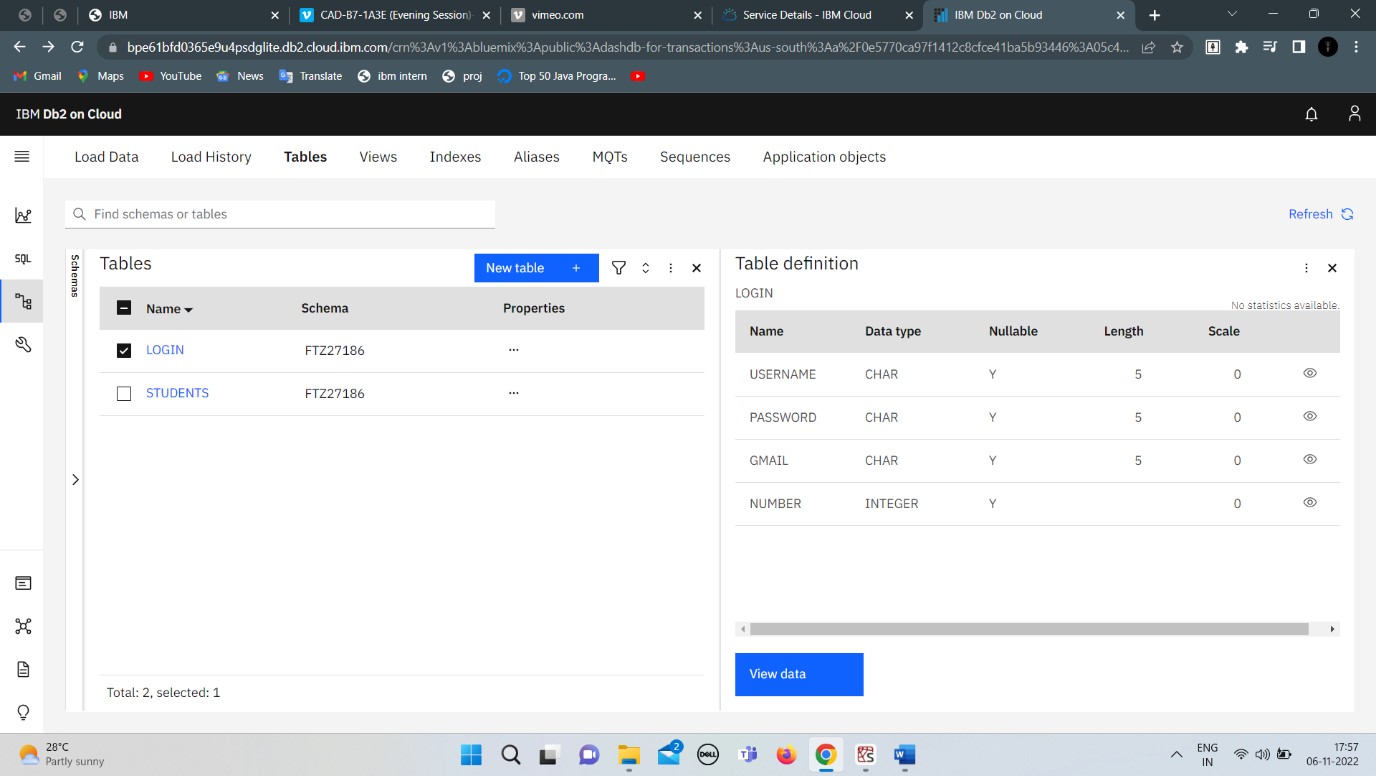
|  |  |
| --- | --- |
| Team ID | PNT2022TMID51155 |
| Project Name | Smart Fashion Recommender Application |

**Create the IBM Db2 service in the IBM cloud and connect the python code with DB.**



# App.py

from flask import Flask,render\_template,request,session import ibm\_db import re app=Flask(\_\_name\_\_) app.secret\_key = 'a'

conn = ibm\_db.connect("DATABASE=bluedb;HOSTNAME=fbd88901-ebdb-4a4f-a32e-

9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=32731;SECURITY=SSL;SSLSE

RVERCERTIFICATE=DigiCertGlobalRootCA.crt;UID=ftz27186;PWD=pwV52WSdXDAHVnLP",'','')

@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'] msg = 'Logged in succesfully !' 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']) def register(): 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 sucsessfully 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\_tempplate('dashboard.html') @app.route('/apply',method =['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" elif request.method == 'POST' : msg = 'Please fill out the form !'

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

@app.route('/display') def display():

print(session["usename"],session[id]) cursor=mysql.connection.cursor() cursor,execute('selet\*from job where userid = 5 s',(session['id'],)) account=cursor.fetchone()

print("accountdisplay",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')