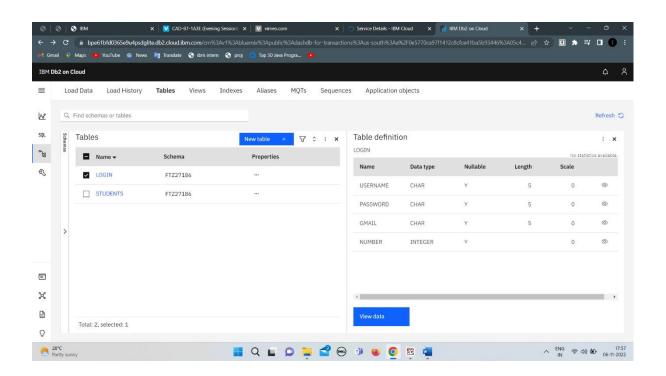| Team ID	PNT2022TMID51155
Project Name	Smart Fashion Recommender Application

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)
```

```python
        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')
```