

Implementing Web Application

IBM DB2 with Python

Date	20 November 2022
Team ID	PNT2022TMID34632
Project Name	Smart Fashion Recommendation System

App.py

```
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=8e359033-a1c9-4643-82ef-8ac06f5107eb.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=30120;SECURITY=SSL;SSLServerCertificate=Dig.crt;", "USERNAME=zdg80277;", "PASSWORD=TvckWLV51123R3Eu;")

@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['Logged'] = True
            session['id'] = account['USERNAME']
```

```

        userid = account['USERNAME']
        session['USERNAME'] = account['USERNAME']
        msg = 'Logged success'
        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():
    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 exists !'
        elif not re.match(r'^@+@[^@]+\.[^@]+',email):
            msg = 'Invalid email id'
        elif not re.match(r'[A-Za-z0-9]+',username):
            msg = 'must only contain character n nos !'
        else:
            insert_sql = 'INSERT INTO users VALUES (?,?,?)'
            prep_stmt = ibm_db.prepare(conn,insert_sql)
            ibm_db.bind_param(prepare_stmt,1,username)
            ibm_db.bind_param(prepare_stmt,2,email)
            ibm_db.bind_param(prepare_stmt,3,password)
            ibm_db.execute(prepare_stmt)
            msg = 'Registration Success !'

    elif request.method == 'POST':
        msg = 'Fill d form !'
    return render_template('register.html',msg = msg)

@app.route('/dashboard')
def dash():
    return render_template('dashboard.html')

```

```

@app.route('/apply',methode = ['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 =?'

        insert_sql = 'INSERT INTO jobs VALUES (?,?,,?,?)'
        prep_stmt = ibm_db.prepare(conn, insert_sql)
        ibm_db.bind_param(prepare_stmt,1,username)
        ibm_db.bind_param(prepare_stmt,2,email)
        ibm_db.bind_param(prepare_stmt,3,qualification)
        ibm_db.bind_param(prepare_stmt,4,skills)
        ibm_db.bind_param(prepare_stmt,5,jobs)
        ibm_db.execute(prepare_stmt)
        msg = 'job applied Successfullf !'

    elif request.method == 'POST':
        msg = "fill d 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 users =%s',(session['id']))
    account = cursor.fetchone()
    print('account display, 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(debug = True)
```

