

Project Name	Smart Fashion Recommender Application
Team ID	PNT2022TMID32347

Create IBM DB2 And Connect With Python

This is simple flask authentication file used for the connection of IBM DB2

```
from flask import Blueprint,render_template,request,redirect,flash,url_for
from werkzeug.security import generate_password_hash, check_password_hash
from src.services import dbServices,otpServices
from src.constants import emailconstant
from . import home
from datetime import datetime
import ibm_db
import re

#IBM_Db2 initiated
conn = dbServices.conn
print(conn)

auth = Blueprint('auth',__name__)

@auth.route("otpGenerate/<fullname>/<mobileNumber>/<email>/<generate>/<sno>")
def otpGenerate(fullname,mobileNumber,email,generate,sno):
    get_otp = otpServices.verifyOTP("+91"+mobileNumber)
    return
    redirect(url_for('auth.processOTP',fullname=fullname,mobileNumber=mobileNumber,email=email,generate=generate,get_otp=get_otp,sno=sno))

@auth.route("signup/",methods=['GET','POST'])
def signup():
    if request.method == 'POST':
        fullname = request.form['name']
        mobileNumber = request.form['phone_number']
        email = request.form['email']
        password = request.form['password']
        confirm_password = request.form['cpassword']
        generate = generate_password_hash(password,method='sha256')
        if confirm_password != password:
            flash('Password is not matched', category='error')
        elif(not re.fullmatch(emailconstant.emailChecker,email)):
            flash('Invalid email format',category='error')
        else:
            check_exist ="SELECT * FROM CUSTOMERDETAILS WHERE MOBILE_NUMBER=? OR EMAIL=?"
            Prep_select = ibm_db.prepare(conn,check_exist)
            ibm_db.bind_param(Prep_select,1,mobileNumber)
            ibm_db.bind_param(Prep_select,2,email)
            ibm_db.execute(Prep_select)
            verify=ibm_db.fetch_assoc(Prep_select)
            if verify:
                flash('Account already exist!', category='error')
            else:
                print(verify['EMAIL'])
```

```

        return
    redirect(url_for('auth.otpGenerate',fullname=fullname,mobileNumber=mobileNumber,email=email,generate=generate,sno=verify["SNO"]))

    return render_template("sign-up.html")

@auth.route("processOTP/<fullname>/<mobileNumber>/<email>/<generate>/<get_otp>/<sno>",methods=['POST','GET'])
def processOTP(fullname,mobileNumber,email,generate,get_otp,sno):
    if request.method == 'POST':
        if request.form['otp'] == get_otp:
            try:
                flash('Account Created!','success')
                insert_sql ="INSERT INTO
CUSTOMERDETAILS(USERNAME,MOBILE_NUMBER,EMAIL,PASSWORD,REGISTERED_ON,LAST_VISITED_AT)
VALUES(?,?,?,?,?,?);"
                prep_stmt = ibm_db.prepare(conn,insert_sql)
                ibm_db.bind_param(prep_stmt,1,fullname)
                ibm_db.bind_param(prep_stmt,2,mobileNumber)
                ibm_db.bind_param(prep_stmt,3,email)
                ibm_db.bind_param(prep_stmt,4,generate)
                ibm_db.bind_param(prep_stmt,5,str(datetime.now()))
                ibm_db.bind_param(prep_stmt,6,str(datetime.now()))
                ibm_db.execute(prep_stmt)
            except Exception as e:
                print(e)
                return redirect(url_for('auth.signup'))
            return
    redirect(url_for('home.user',username=fullname,email=email,mobile=mobileNumber,sno=sno))
    else:
        flash('OTP is not matched','error')

    return
    render_template("otp.html",fullname=fullname,mobileNumber=mobileNumber,email=email,generate=generate)

@auth.route("signin/",methods=['GET','POST'])
def signin():

    if request.method=='POST':
        uname=request.form['email']
        password=request.form['password']
        if(uname.isdigit()):
            if(len(uname)==10):
                checkpass = "SELECT * FROM CUSTOMERDETAILS WHERE MOBILE_NUMBER=?"
                check_password = ibm_db.prepare(conn,checkpass)
                ibm_db.bind_param(check_password,1,uname)
                ibm_db.execute(check_password)
                verifyPass = ibm_db.fetch_assoc(check_password)
                if verifyPass:
                    if check_password_hash(verifyPass["PASSWORD"],password):
                        flash('Welcome {0}'.format(verifyPass["USERNAME"]), 'success')
                        update_sql ="UPDATE CUSTOMERDETAILS SET LAST_VISITED_AT=? WHERE
MOBILE_NUMBER=? AND PASSWORD=?"
                        prep_stmt = ibm_db.prepare(conn,update_sql)
                        ibm_db.bind_param(prep_stmt,1,datetime.now())

```

```

        ibm_db.bind_param(prepare_stmt,2,verifyPass["MOBILE_NUMBER"])
        ibm_db.bind_param(prepare_stmt,3,verifyPass["PASSWORD"])
        ibm_db.execute(prepare_stmt)
        return
    redirect(url_for("home.user",username=verifyPass["USERNAME"],email=verifyPass["EMAIL"],mobile=verifyPass["MOBILE_NUMBER"],sno=verifyPass["SNO"]))

    else:
        flash('Invalid Password!',category='error')
    else:
        flash('Invalid Mobile number!',category='error')
    else:
        flash('Invalid Mobile number!',category='error')
    elif(re.fullmatch(emailconstant.emailChecker,uname)):
        checkpass = "SELECT * FROM CUSTOMERDETAILS WHERE EMAIL=?"
        check_password = ibm_db.prepare(conn,checkpass)
        ibm_db.bind_param(check_password,1,uname)
        ibm_db.execute(check_password)
        verifyPass = ibm_db.fetch_assoc(check_password)
        if verifyPass:
            if check_password_hash(verifyPass["PASSWORD"],password):
                flash('Welcome {0}'.format(verifyPass["USERNAME"]), 'success')
                update_sql ="UPDATE CUSTOMERDETAILS SET LAST_VISITED_AT=? WHERE EMAIL=? AND PASSWORD=?"
                prepare_stmt = ibm_db.prepare(conn,update_sql)
                ibm_db.bind_param(prepare_stmt,1,datetime.now())
                ibm_db.bind_param(prepare_stmt,2,verifyPass["EMAIL"])
                ibm_db.bind_param(prepare_stmt,3,verifyPass["PASSWORD"])
                ibm_db.execute(prepare_stmt)
                return
    redirect(url_for("home.user",username=verifyPass["USERNAME"],email=verifyPass["EMAIL"],mobile=verifyPass["MOBILE_NUMBER"],sno=verifyPass["SNO"]))
    else:
        flash('Invalid Password!',category='error')
    else:
        flash('Invalid Email address!',category='error')

    else:
        flash('Invalid email and password!',category='error')

    return render_template("sign-in.html")

@auth.route("forget/",methods=['POST','GET'])
def forget():
    if request.method == "POST":
        uname = request.form['email']
        if(uname.isdigit()):
            if(len(uname)==10):
                checkpass = "SELECT * FROM CUSTOMERDETAILS WHERE MOBILE_NUMBER=?"
                check_password = ibm_db.prepare(conn,checkpass)
                ibm_db.bind_param(check_password,1,uname)
                ibm_db.execute(check_password)
                verifyPass = ibm_db.fetch_assoc(check_password)
                if verifyPass:
                    return
    redirect(url_for("auth.change",mobilenumber=verifyPass["MOBILE_NUMBER"]))
    else:

```

```

        flash("Account doesn't exist",category='error')
    else:
        flash('Invalid Mobile number!',category='error')
    else:
        flash('Invalid Mobile number!',category='error')
    return render_template("forget.html")

#balance work in authenticaiton:
#visible icon ok
#session
#cookie

@auth.route("change/<mobilenumber>",methods=['POST','GET'])
def change(mobilenumber):
    if request.method == "POST":
        new_password = request.form['new_password']
        c_password = request.form['c_password']

        if(new_password == c_password):
            get_otp = otpServices.verifyOTP("+91"+mobilenumber)
            return
    redirect(url_for("auth.processOTP_FORGET",mobileNumber=mobilenumber,get_otp=get_otp,password=new_password))
    else:
        flash('Password is not matched',category='error')
    return render_template("change.html",msg="none")

@auth.route("processFORGET/<mobileNumber>/<get_otp>/<password>",methods=['POST','GET'])
def processOTP_FORGET(password,mobileNumber,get_otp):
    if request.method == 'POST':
        if request.form['otp'] == get_otp:
            try:
                flash('Password Changed!','success')
                update_sql ="UPDATE CUSTOMERDETAILS SET PASSWORD = ? WHERE
MOBILE_NUMBER= ?"
                prep_stmt = ibm_db.prepare(conn,update_sql)
                ibm_db.bind_param(prep_stmt,1,generate_password_hash(password,method='sha256'))
                ibm_db.bind_param(prep_stmt,2,mobileNumber)
                ibm_db.execute(prep_stmt)
                return render_template('change.html',msg="success")
            except:
                flash('Problem with internal system')
        else:
            flash('OTP is not matched','error')

    return render_template("otp.html")

@auth.route("logout/")
def logout():
    return redirect(url_for("home.index"))

```

