

WEB UI CONNECT TO THE IBM DB2

```
from flask import Flask,render_template,request,session,logging,url_for,redirect,flash
from sqlalchemy import create_engine
from sqlalchemy.orm import scoped_session,sessionmaker
from passlib.hash import sha256_crypt

engine = create_engine("mysql+pymysql://root:sabi@localhost/register")
#{mysql+pysql://username:password@localhost/databasename)
db=scoped_session(sessionmaker(bind=engine))

app=Flask(__name__)

@app.route("/")
def home():
    return render_template("home.html")

#register form

@app.route("/register",methods=["GET","POST"])
def register():
    if request.method == "POST":
        name= request.form.get("name")
        email =request.form.get("email")
        password=request.form.get("password")
        confirm=request.form.get("confirm")
        secure_password = sha256_crypt.encrypt(str(password))
        if password == confirm:
            db.execute("Insert into users(name,email,password)
values(:name,:email,:password)","{name":name,"email":email,"password":secure_password})
            db.commit()
            flash("you are registered and can login","Success")
            return redirect(url_for('login'))
```

```

else:
    flash("password does not match","danger")
    return render_template("register.htm")
    return render_template("register.htm")

#login
@app.route("/login",methods=["GET","POST"])
def login():
    if request.method == "POST":
        name = request.form.get("name")
        password = request.form.get("password")
        namedata = db.execute("SELECT name FROM users WHERE
name=:name",{"name":name}).fetchone()
        passworddata = db.execute("SELECT password FROM users WHERE
name=:name",{"name":name}).fetchone()

        if namedata is None:
            flash("No username","danger")
            return render_template("login.htm")
        else:
            for passwor_data in passworddata:
                if sha256_crypt.verify(password,passwor_data):
                    session["log"] = True

            flash("You are now login","success")
            return redirect(url_for('photo'))
        else:
            flash("incorrect password","danger")
            return render_template("login.htm")

```

```
    return render_template("login.htm")

#photo
@app.route("/photo")
def photo():
    return render_template("photo.htm")

#logout
@app.route("/logout")
def logout():
    session.clear()
    flash("OOOOOOH You have been logged off","Credited")
    return redirect(url_for('login'))
```