1. Create User table with user with email, username, roll number, password.

2. Perform UPDATE ,DELETE Queries with user table

3. Connect python code to db2.

4. Create a flask app with registration page, login page and welcome page. By default load the registration page once the user enters all the fields store the data in database and navigate to login page authenticate user

username and password. If the user is valid show the welcome page .

Apply.html

<html>

<body>

<form action="http://localhost:5000/login" method="POST">

<p>Enter Username:</p>

<p><input type="text" name="username" /></p>

<p>Enter Email:</p>

<p><input type="email" name="email" /></p>

<p>Enter Qualification :</p>

<p><input type="text" name="qualification" /></p>

<p>Enter Skill:</p>

<p><input type="text" name="skill" /></p>

<p>Enter Jobs:</p>

<p><input type="text" name="jobs" /></p>

<p><input type="submit" value="submit"></p>

</form>

</body>

</html>

Login.html

<html>

<body>

<form action="http://localhost:5000/login" method="POST">

<p>Enter Username:</p>

<p><input type="text" name="username" /></p>

<p>Enter Password :</p>

<p><input type="password" name="password" /></p>

<p><input type="submit" value="submit"></p>

</form>

</body>

</html>

Register.html

<html>

<body>

<form action="http://localhost:5000/login" method="POST">

<p>Enter Username:</p>

<p><input type="text" name="username" /></p>

<p>Enter Email:</p>

<p><input type="email" name="email" /></p>

<p>Enter Password :</p>

<p><input type="password" name="password" /></p>

<p><input type="submit" value="submit"></p>

</form>

</body>

</html>

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=b70af05b-76e4-4bca-a1f5-

23dbb4c6a74e.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=32716;SECURITY=SSL;SSLServerC ertificate=DigiCertGlobalRootCA.crt;UID=jzc43091;PWD=PI8VtGRvZlSVT65A ",'','')

@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']

userid= account['USERNAME']

session['username'] = account['USERNAME']

msg = 'Logged in successfully !'

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 registet():

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 successfully 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 template('dashboard.html')

@app.route('/apply',methods =['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('apply.html', msg = msg)

@app.route('/display')

def display():

print(session["username"],session['id'])

cursor = mysql.connection.cursor()

cursor.execute('SELECT \* FROM job WHERE userid = % s', (session['id'],))

account = cursor.fetchone()

print("accountdislay",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](https://0.0.0.0)')