

IMPLEMENTING WEB APPLICATION

Create IBM DB2 and Connect with Python

Team ID	PNT2022TMID46377
Project Name	Skill And Job Recommender Application

Code:

```
import Flask, render_template, request, redirect, url_for, session

import ibm_db
import bcrypt
conn=
ibm_db.connect("DATABASE=bludb;HOSTNAME=;PORT=;SECURITY=SSL;SSLServerCertificate=
DigiCertGlobalRootCA.crt;UID=;PWD='', '')

# url_for('static', filename='style.css')

app = Flask(__name__)
app.secret_key=b'_5#y2L"F4Q8z\xn\xec]/'

@app.route("/", methods=['GET'])
def home():
    if 'email' not in session:
        return redirect(url_for('login'))
    return render_template("home.html", name='Home')

@app.route("/register", methods=['GET', 'POST'])
def register():
    if request.method == 'POST':
        email=request.form['email']
        username=request.form['username']
        rollNo=request.form['rollNo']
        password=request.form['password']

        if not email or not username or not rollNo or not password:
            return render_template("register.html", error='Please fill all fields')

        hash=bcrypt.hashpw(password.encode('utf-8'),bcrypt.gensalt())
```

```

query="SELECT*FROMUSERWHEREEmail=?ORrollNo=?" stmt =
ibm_db.prepare(conn, query)
ibm_db.bind_param(stmt,1,email)
ibm_db.bind_param(stmt,2,rollNo)
ibm_db.execute(stmt)
isUser = ibm_db.fetch_assoc(stmt)

if not isUser:
    insert_sql = "INSERT INTO User(username,email,PASSWORD,rollNo) 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,hash)
    ibm_db.bind_param(prepare_stmt,4,rollNo)
    ibm_db.execute(prepare_stmt)
    returnrender_template('register.html',success="Youcanlogin")
else:
    returnrender_template('register.html',error='InvalidCredentials')

returnrender_template('register.html',name='Home')

@app.route("/login",methods=['GET','POST']) def
login():
    ifrequest.method=="POST":
        email=request.form['email']
        password = request.form['password']

        ifnotemailornotpassword:
            returnrender_template('login.html',error='Pleasefillallfields')
        query = "SELECT * FROM USER WHERE email=?"
        stmt = ibm_db.prepare(conn, query)
        ibm_db.bind_param(stmt,1,email)
        ibm_db.execute(stmt)
        isUser = ibm_db.fetch_assoc(stmt)
        print(isUser,password)

        if not isUser:
            returnrender_template('login.html',error='InvalidCredentials')

        isPasswordMatch = bcrypt.checkpw(password.encode('utf
- 8'),isUser['PASSWORD'].encode('utf-8'))

        if not isPasswordMatch:

```

```

        return render_template('login.html', error='InvalidCredentials')

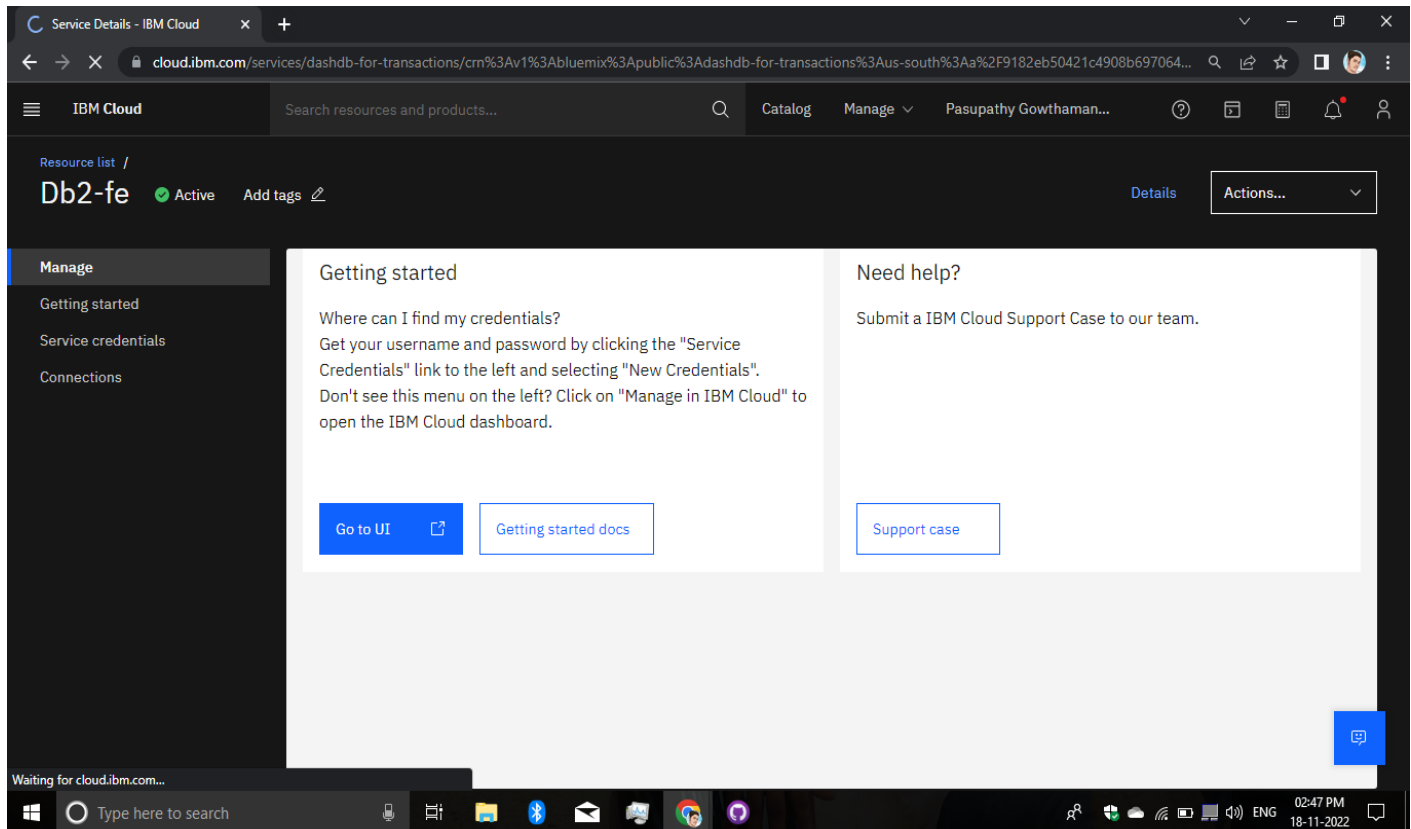
    session['email'] = isUser['EMAIL']
    return redirect(url_for('home'))

    return render_template('login.html', name='Home')

@app.route('/logout')
def logout():
    session.pop('email', None)
    return redirect(url_for('login'))

```

Outputs:



Service Details - IBM Cloud

cloud.ibm.com/services/dashdb-for-transactions/cm%3Av1%3Abluemix%3Apublic%3Adashdb-for-transactions%3Aus-south%3Aa%2F9182eb50421c4908b697064...

IBM Cloud

Search resources and products...

Catalog Manage Pasupathy Gowthaman...

Resource list / **Db2-fe** Active Add tags

Details Actions...

Manage

Getting started

Service credentials

Connections

Service credentials

You can generate a new set of credentials for cases where you want to manually connect an app or external consumer to an IBM Cloud service. [Learn more](#)

Search credentials...

New credential

Key name	Date created
Service credentials-1	2022-11-18 12:10 PM

Type here to search

ENG 02:51 PM 18-11-2022

Service Details - IBM Cloud

IBM Db2 on Cloud

bpe61bfd0365e9u4psdglite.db2.cloud.ibm.com/cm%3Av1%3Abluemix%3Apublic%3Adashdb-for-transactions%3Aus-south%3Aa%2F9182eb50421c4908b69706424142...

IBM Db2 on Cloud

Overview In-flight executions Connections Table performance

Resource usage

Last 1 hour

Storage (10M / 200M)

5% current value

Storage usage (%)

Time

01:28 PM 01:40 PM 02:00 PM 02:28 PM

Type here to search

ENG 02:54 PM 18-11-2022