Create IBM DB2 and Connect with Python

Team ID	PNT2022TMID46366
Project Name	News Tracker Application

Create IBM DB2 and Connect with Python:

```
from flask import Flask, render_template, request
import ibm_db
import bcrypt
from dotenv import load_dotenv
import os

load_dotenv()

db = os.getenv("DATABASE")
host = os.getenv("HOSTNAME")
port = os.getenv("PORT")
sslcert = os.getenv("SSLServerCertificate")
userId = os.getenv("UID")
password = os.getenv("PWD")
print(db,port)

conn =
ibm_db.connect(f'DATABASE={db};HOSTNAME={host};PORT={p
ort};SECURITY=SSL;SSLServerCertificate={sslcert};UID={userId};P
WD={password}','','')
```

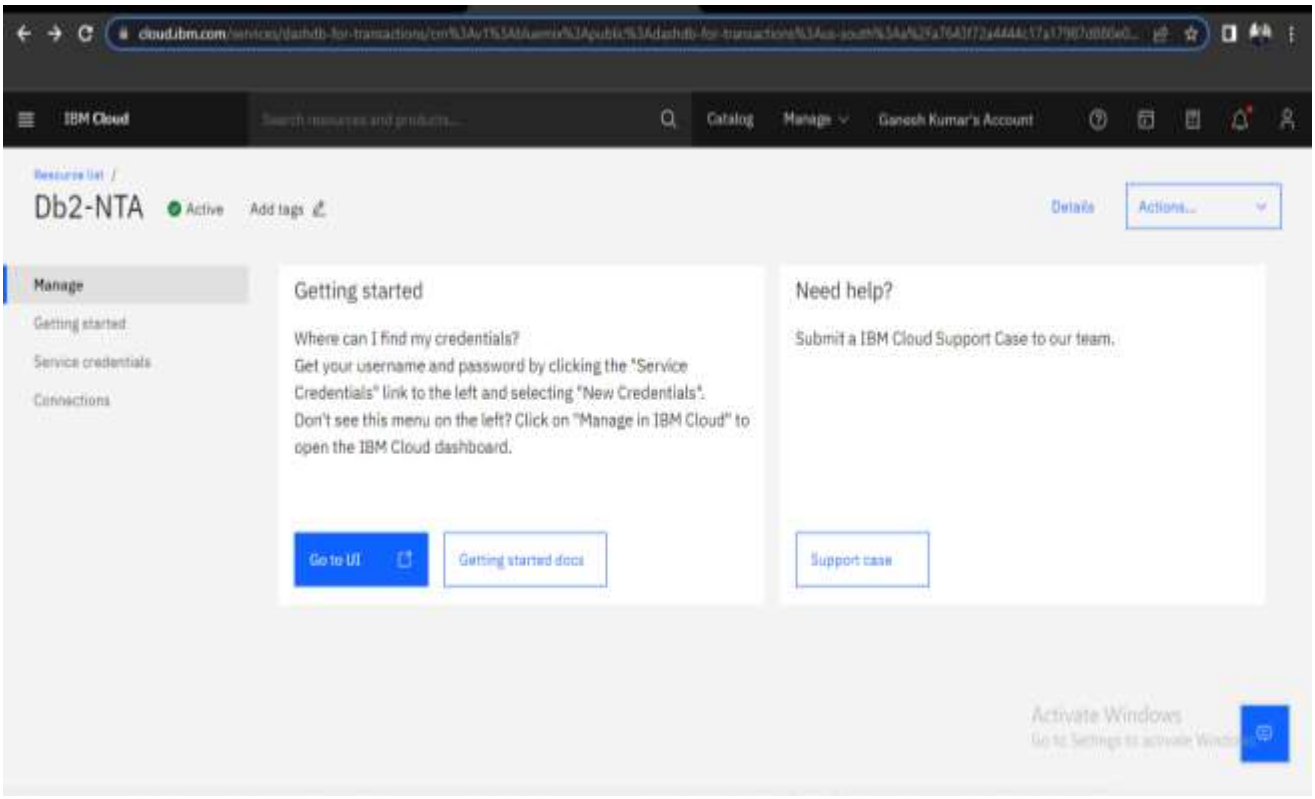
```
app = Flask(__name__)

@app.route('/')
def index():
    return render_template('index.html', title='Home')


@app.route('/about')
def about():
    return render_template('about.html', title='About')


@app.route('/signin', methods=['GET', 'POST'])
def signin():
    if request.method == 'POST':

        email = request.form['email']
        pwd = request.form['password']

        sql = "SELECT password FROM users WHERE email =?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt, 1, email)
        ibm_db.execute(stmt)
        auth_token = ibm_db.fetch_assoc(stmt)
        print("auth",auth_token)

        if auth_token:
            # encoding user password
            userBytes = pwd.encode('utf-8')
```

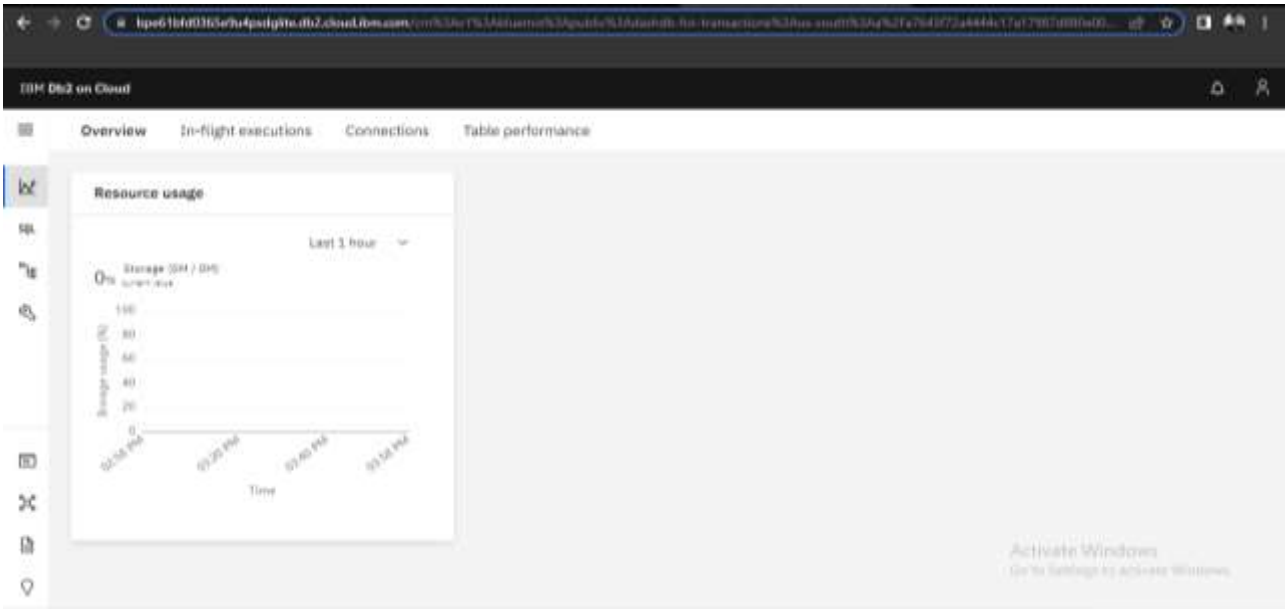
```
        byte_pwd = bytes(auth_token['PASSWORD'], 'utf-8')

        # checking password
        result = bcrypt.checkpw(userBytes, byte_pwd)

        if result:
            print("succ")
            return render_template('index.html', succ="Logged in
Successfully")
        else:
            return render_template('signin.html', fail="Invalid
Credentials")
    else:
        return render_template('signup.html', fail="User doesn't
exist, Please Register using your details!")
  else:
    return render_template('signin.html', title='Sign In')

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

    sql = "SELECT * FROM users WHERE email =?"
    stmt = ibm_db.prepare(conn, sql)
    ibm_db.bind_param(stmt,1,email)
    ibm_db.execute(stmt)
```

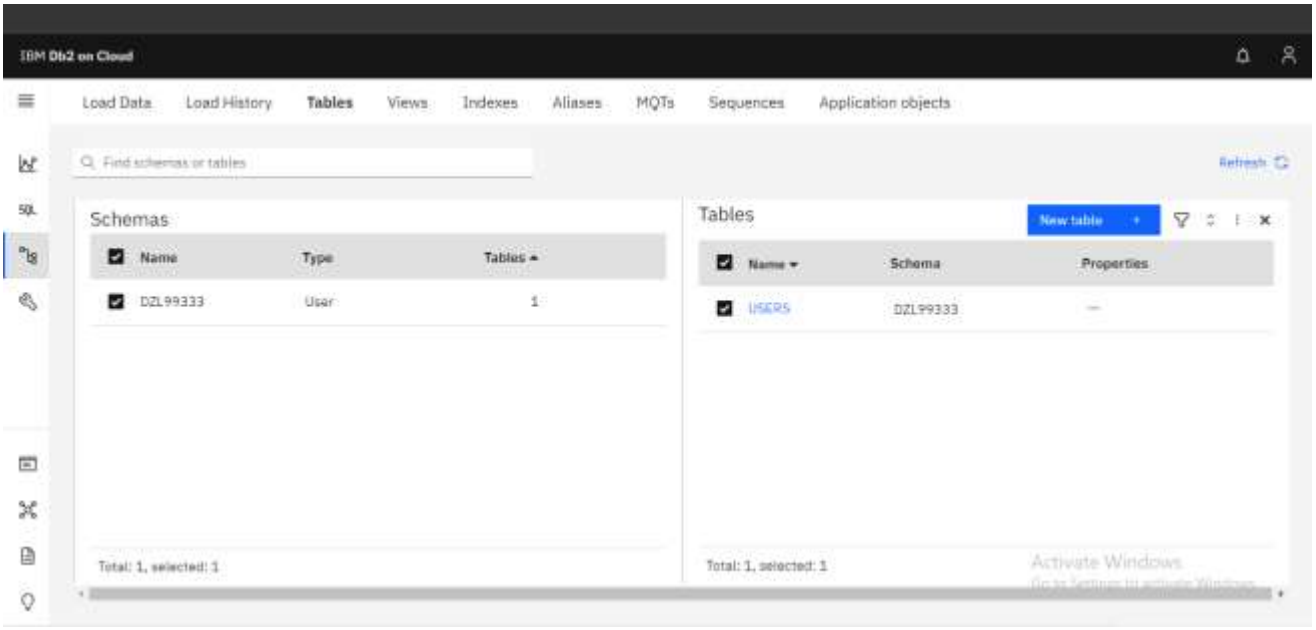
```
        account = ibm_db.fetch_assoc(stmt)

        # converting password to array of bytes
        bytes = password.encode('utf-8')

        # generating the salt
        salt = bcrypt.gensalt()

        # Hashing the password
        hashed_password = bcrypt.hashpw(bytes, salt)

        password = hashed_password

        if account:
            return render_template('signin.html', msg="You are
already a member, please login using your details")
        else:
            insert_sql = "INSERT INTO users (username, password,
name, email) VALUES (?,?,?,?)"
            prep_stmt = ibm_db.prepare(conn, insert_sql)
            ibm_db.bind_param(prep_stmt, 1, username)
            ibm_db.bind_param(prep_stmt, 2, password)
            ibm_db.bind_param(prep_stmt, 3, name)
            ibm_db.bind_param(prep_stmt, 4, email)
            ibm_db.execute(prep_stmt)

            return render_template('index.html', title="Home",
succ="Registration Successfull!")
```

```
    return render_template('signup.html', title='Sign Up')

if __name__ == "__main__":
    app.run(debug=True)
```

Outputs:




IBM Db2 on Cloud
Overview
In-flight executions
Connections
Table performance
Resource usage
Last 1 hour
Time


IBM Db2 on Cloud
Load Data
Load History
Tables
Views
Indexes
Aliases
MQTs
Sequences
Application objects
Find schemas or tables
Refresh
Schemas
Name
Type
Tables
DZL99333
User
1
Total: 1, selected: 1
Tables
New table
Name
Schema
Properties
USERS
DZL99333
Total: 1, selected: 1