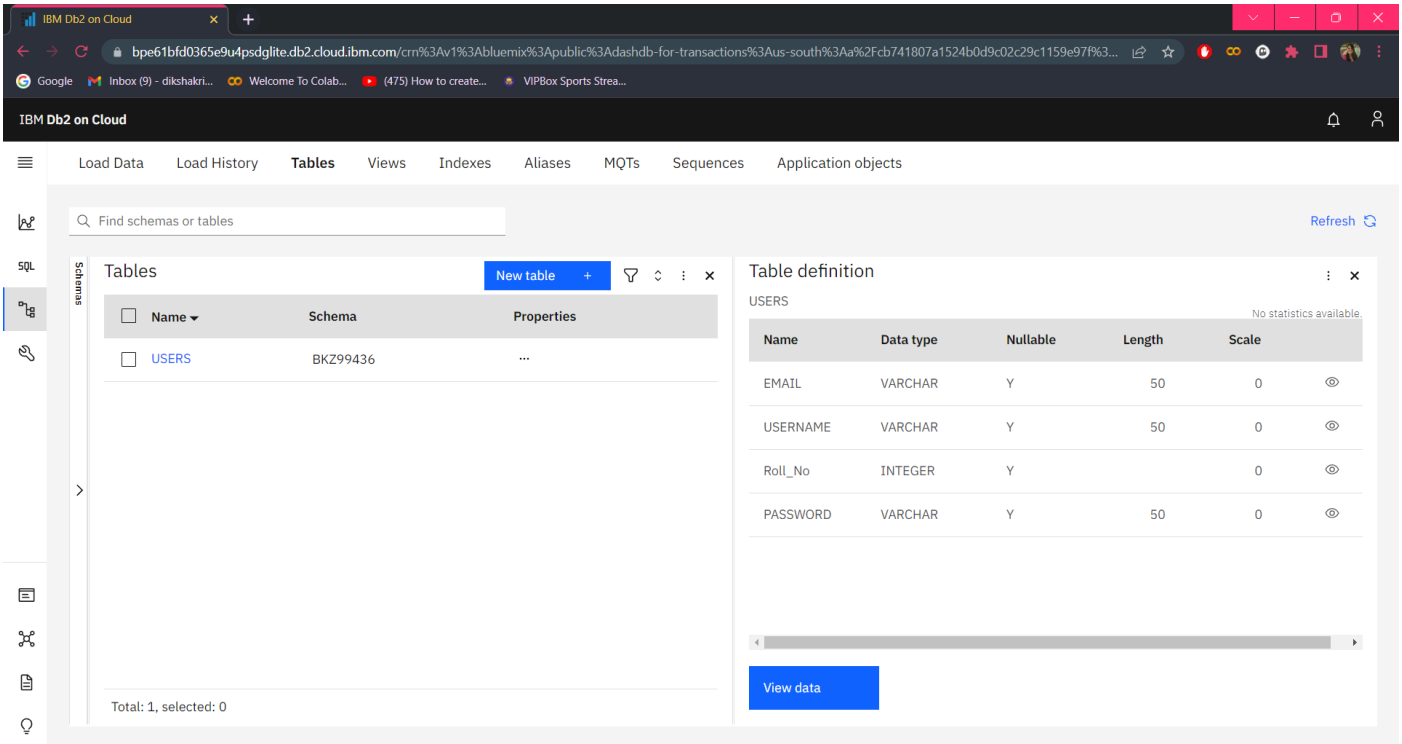
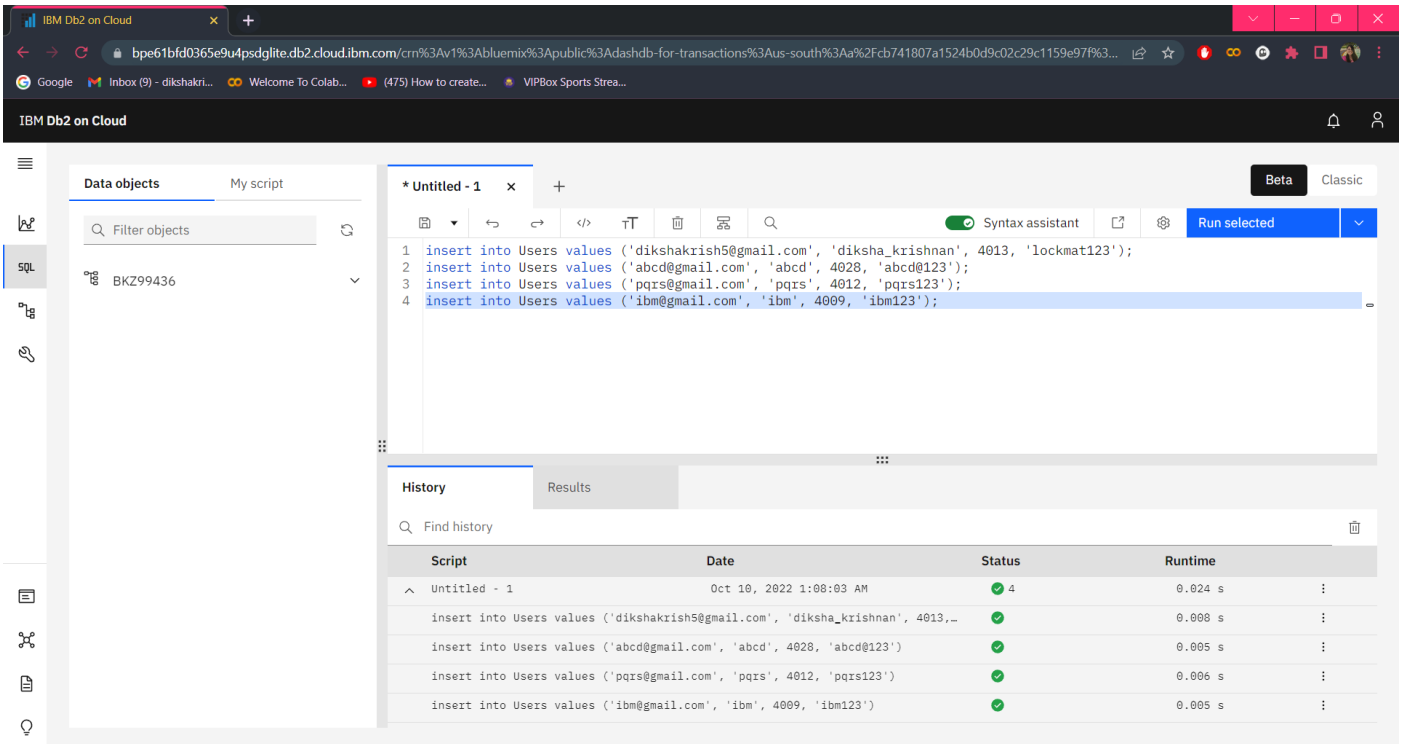
ASSIGNMENT – 2

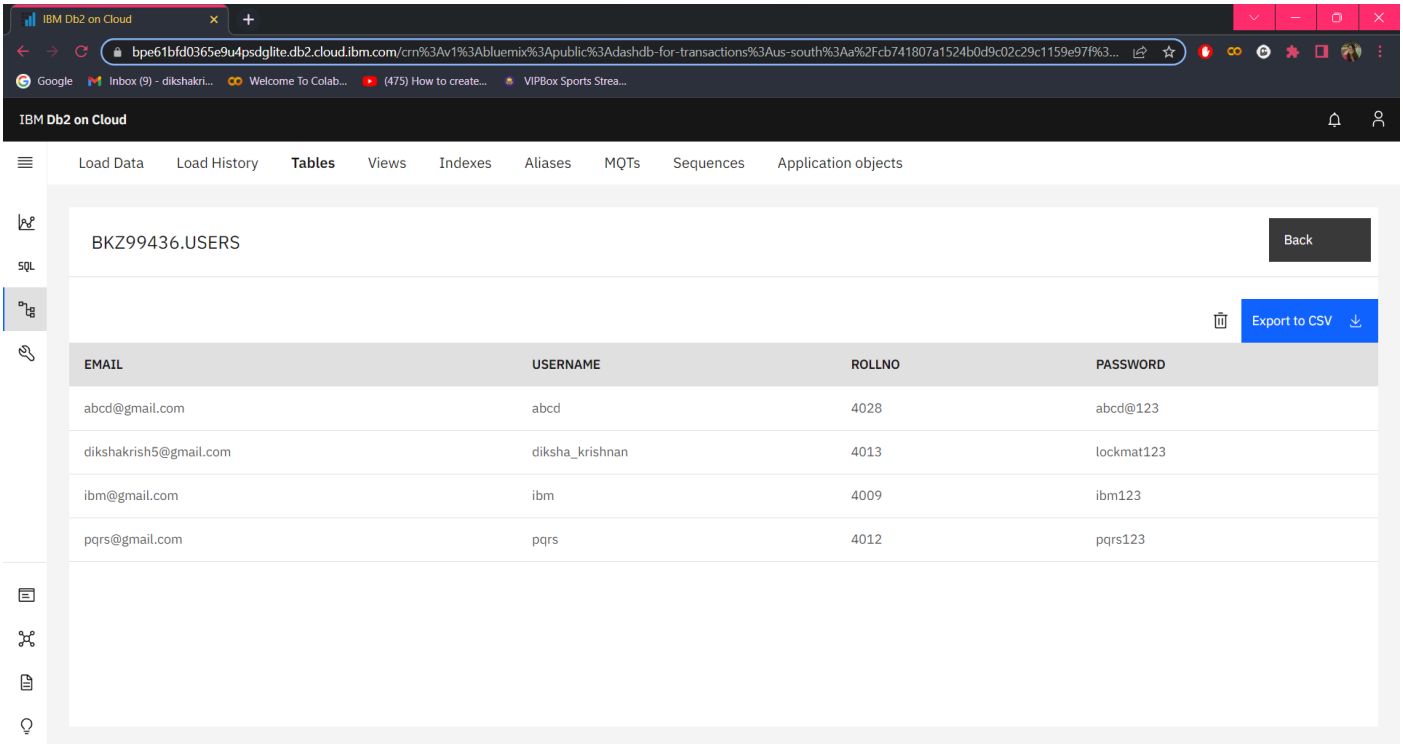
DIKSHA KRISHNAN
311519104015
dikshakrish5@gmail.com
PNT2022TMID27807

1. CREATE USER TABLE WITH USER WITH EMAIL, USERNAME, ROLL NUMBER, PASSWORD

TABLE CREATION:

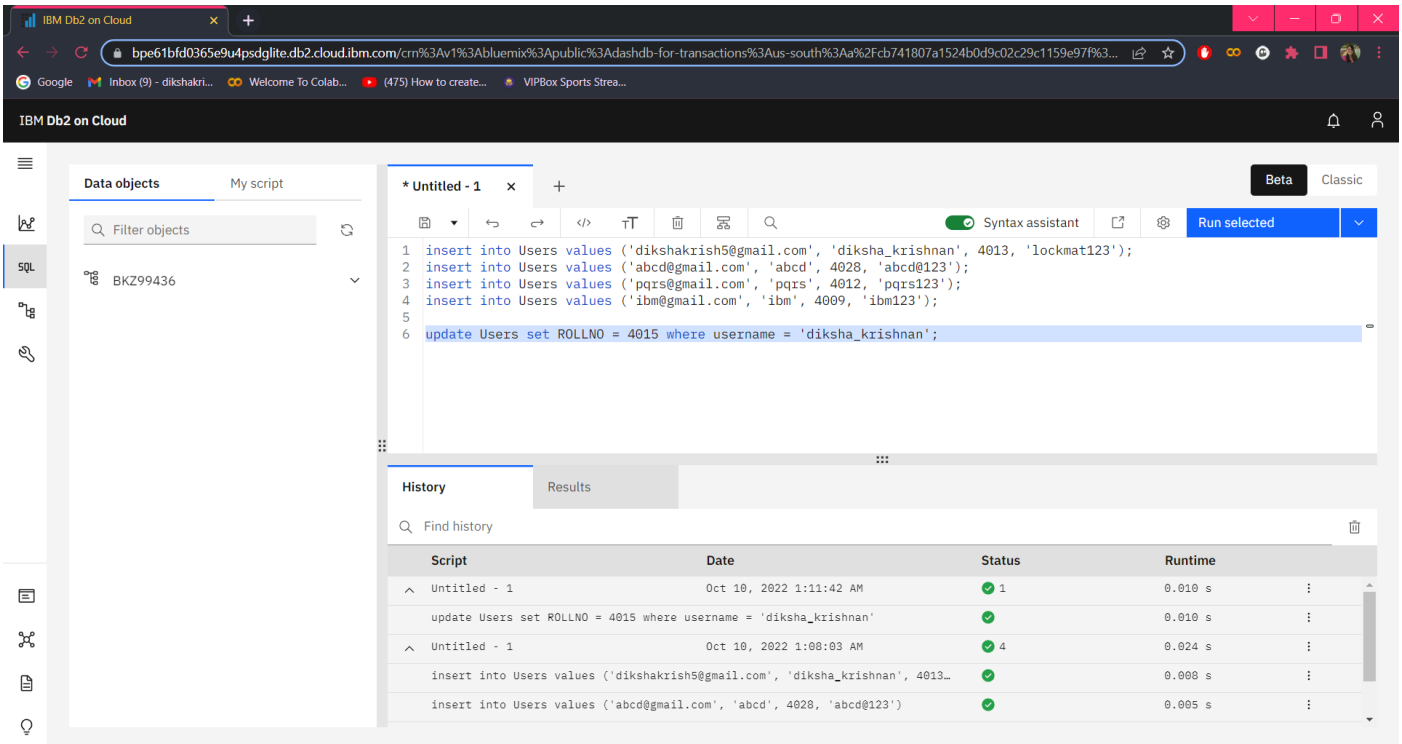


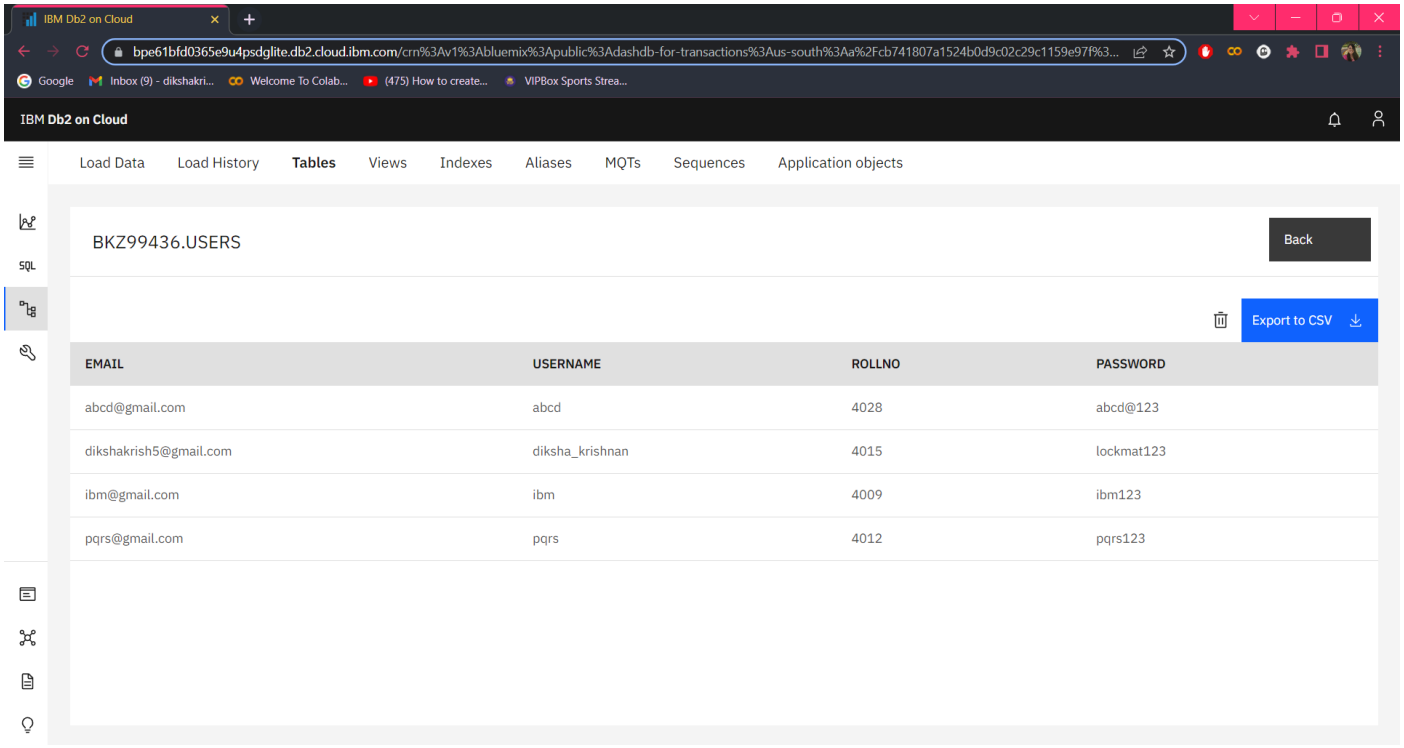




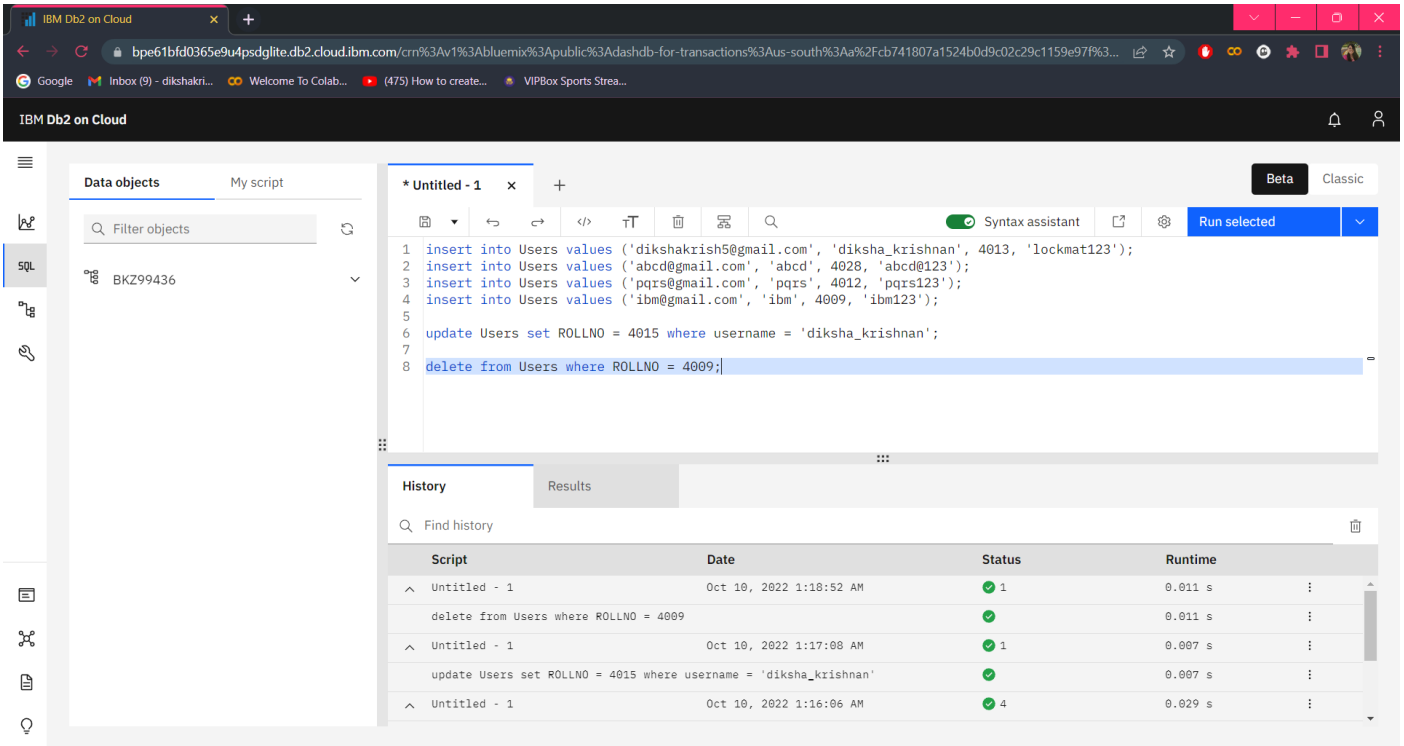
2. PERFORM UPDATE, DELETE QUERIES WITH USER TABLE

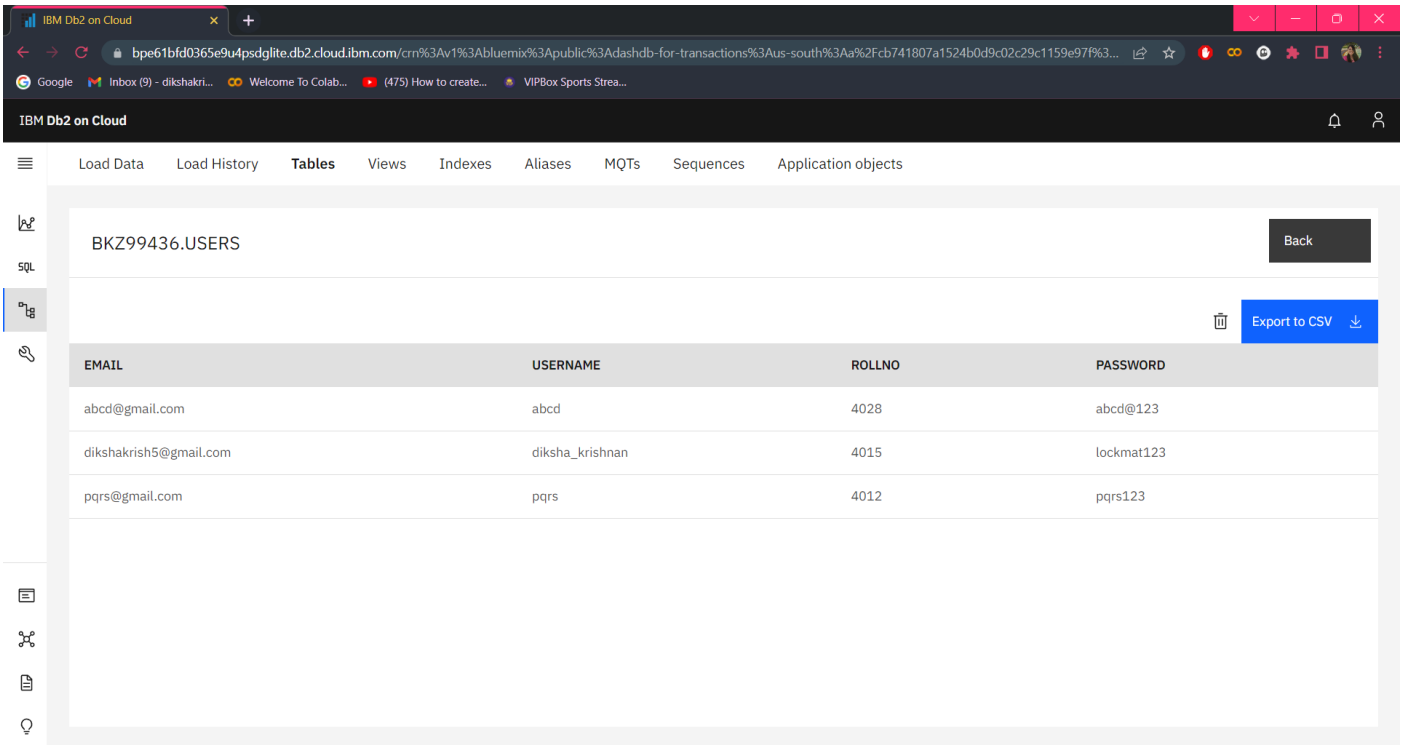
UPDATE QUERY:





DELETE QUERY:





3. CONNECT PYTHON TO DB2

CONNECTING TO DB:

reg2.html (Registration Page):-

```
<html>
<body>
   <center>
      <h3><b>REGISTRATION</b></h3>
      <form action = "http://localhost:5000/reg2" method ="POST">
         <p>Enter Register No.: <input type = "text" name="rollno"/></p>
         <p>Enter Email ID: <input type = "text" name="email"/></p>
         <p>Enter Username: <input type = "text" name="uname"/></p>
         <p>Enter Password: <input type = "password" name="pwd"/></p>
         <p><input type = "submit" value="SUBMIT"/></p>
      </form>
   </center>
</body>
</html>
```

login.html (Login Page):-

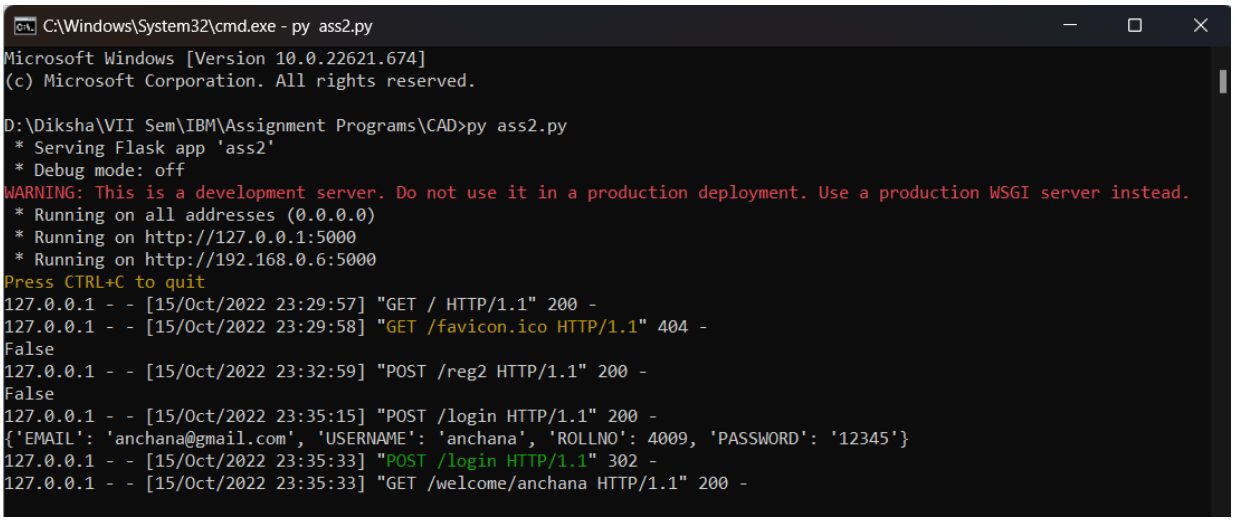
```
<html>
<body>
   <center>
      <h3><b>LOGIN</b></h3>
      <form action = "http://localhost:5000/login" method = "POST">
         <p>Enter Username: <input type = "text" name="uname"/></p>
         <p>Enter Password: <input type = "password" name="pwd"/></p>
         <p><input type = "submit" value="SUBMIT"/></p>
      </form>
   </center>
```

```
</body>
</html>
```

ass2.py:-

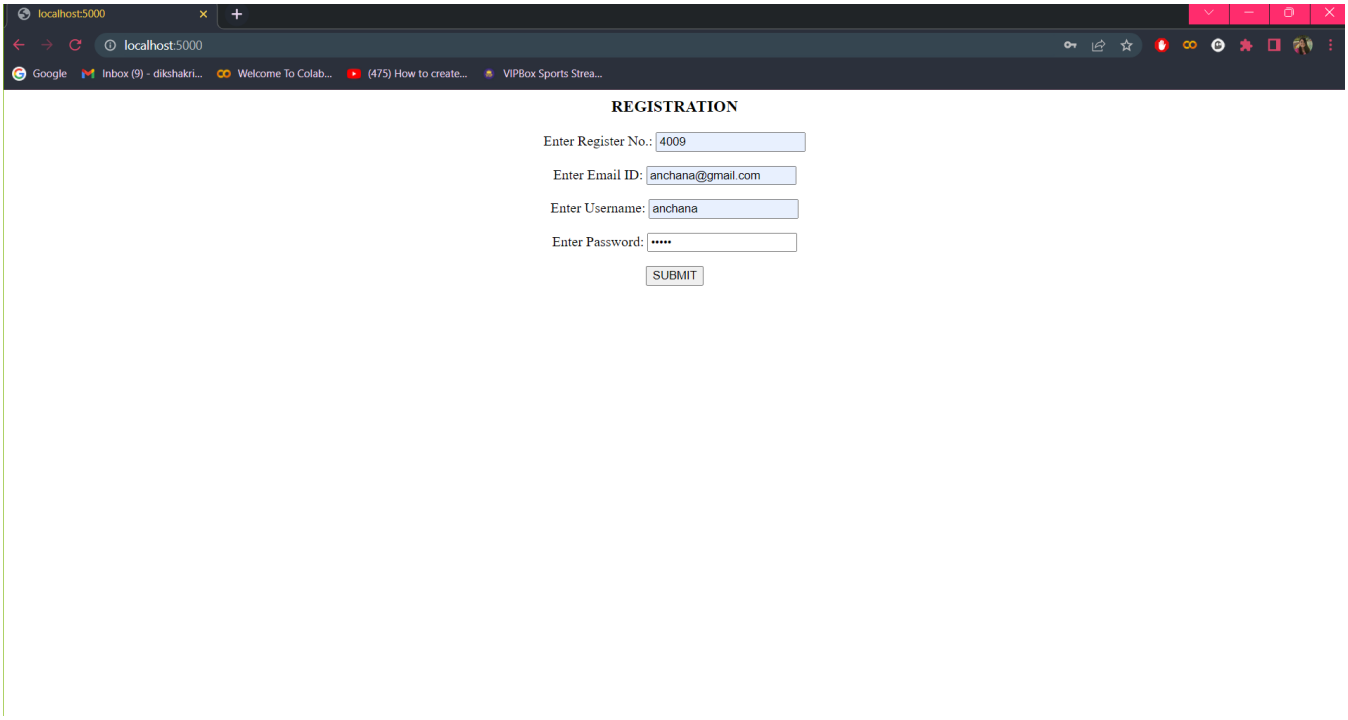
```
from flask import Flask,render_template, redirect, url_for, request, session
import ibm_db
import re

app=Flask(__name__)
app.secret_key='a'

conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=98538591-7217-4024-b027-
8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud;PORT=30875;SECURITY=SSL;SSLSe
rverCertificate=DigiCertGlobalRootCA.crt;UID=bkz99436;PWD=w4b5WZw6Dj9eVXLB;","")

@app.route('/')
def home():
return render_template('reg2.html')

@app.route('/login',methods=["GET","POST"])
def login():
    global userid
    msg=" "

    if request.method=="POST":
        username = request.form['uname']
        password = request.form['pwd']
        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!'
            return redirect(url_for('welcome', username=username))
        else: msg = "Incorrect Username/Password"
    return render_template('login.html', msg = msg)

@app.route('/reg2',methods=["GET","POST"])
def registration():
    msg = " "

    if request.method=="POST":
        username = request.form['uname']
        email = request.form['email']
```

```
        password = request.form['pwd']
        rollno = request.form['rollno']
        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 = "Username must contain only alphabets and numbers."
        else:
            insert_sql = "INSERT INTO USERS VALUES(?,?,?,?)"
            prep_stmt = ibm_db.prepare(conn,insert_sql)
            ibm_db.bind_param(prep_stmt,1,email)
            ibm_db.bind_param(prep_stmt,2,username)
            ibm_db.bind_param(prep_stmt,3,rollno)
            ibm_db.bind_param(prep_stmt,4,password)
            ibm_db.execute(prep_stmt)
            msg = "You have successfully registered."
            return render_template('login.html', msg=msg)

    elif request.method == 'POST': msg="Please fill out the form."
    return render_template('reg2.html',msg=msg)

@app.route('/welcome/<username>')
def welcome(username):
    return "Welcome %s!" %username

if __name__=="__main__":
    app.run(host='0.0.0.0')
```

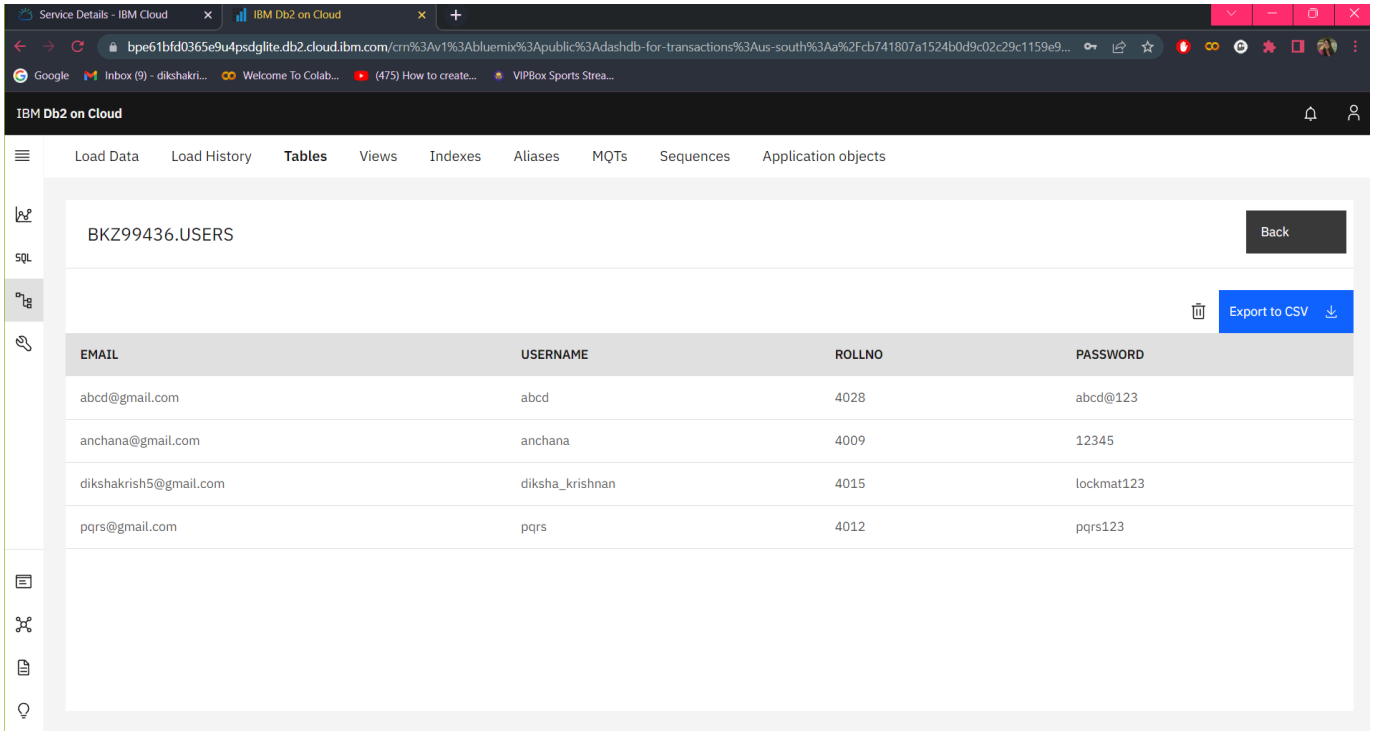


4. CREATE A FLASK APP WITH REGISTRATION PAGE, LOGIN PAGE AND WELCOME PAGE. BY DEFAULT LOAD THE REGISTRATION PAGE ONCE THE USER ENTERS ALL THE FIELDS STORE THE DATA IN DATABASE AND NAVIGATE TO LOGIN PAGE AUTHENTICATE USER USERNAME AND PASSWORD. IF THE USER IS VALID SHOW THE WELCOME PAGE

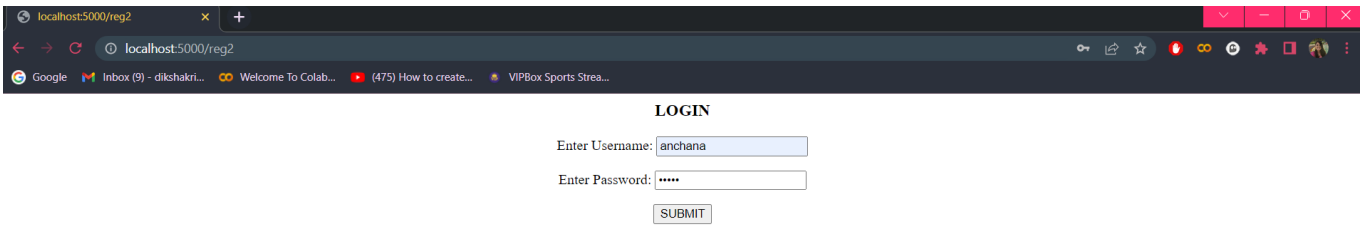
REGISTRATION PAGE:



DATABASE INSERTION:



LOGIN PAGE:



WELCOME PAGE: