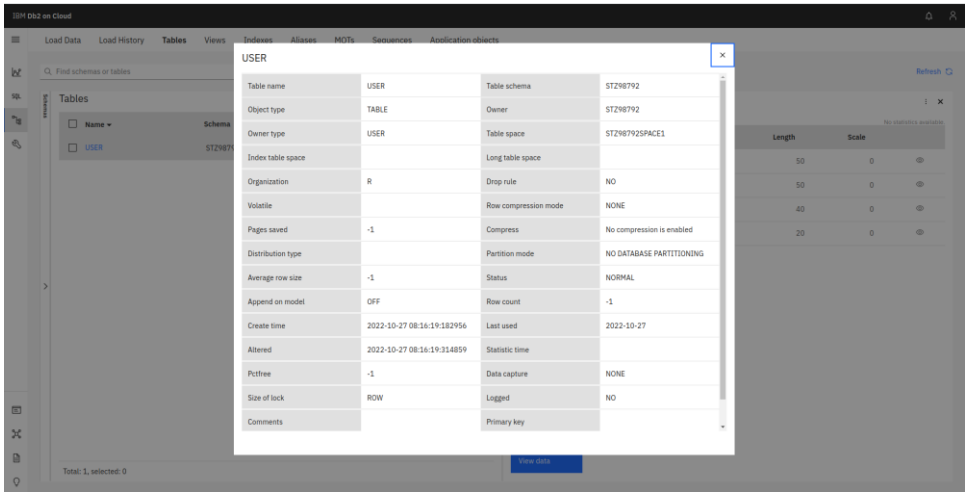
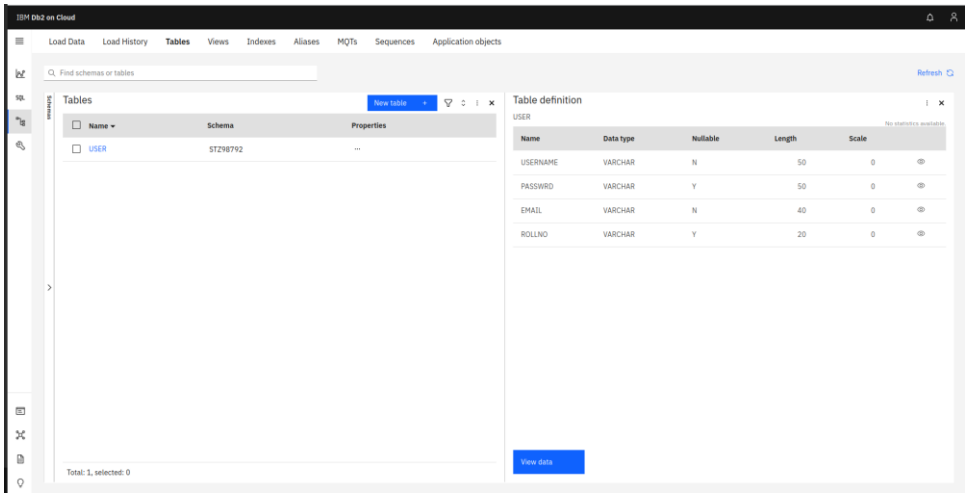
Assignment 2

Raghunandhan AJ

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

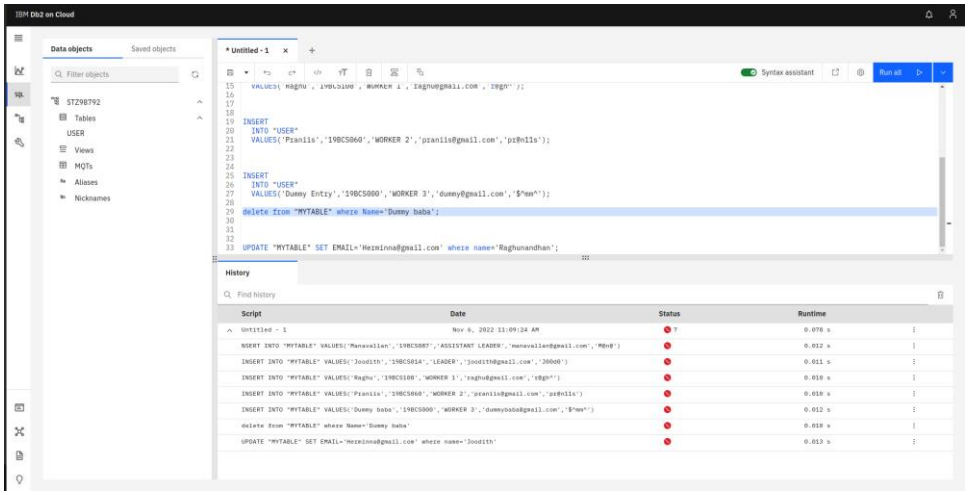
Original table

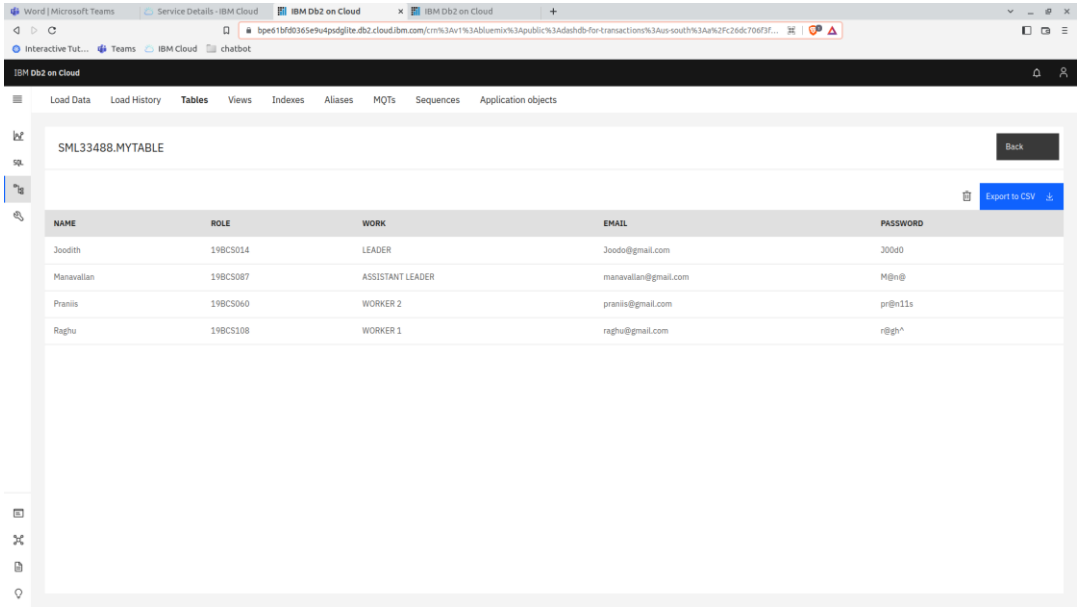




2. Perform UPDATE DELETE Queries with user table

After update and delete





SQL:

```
INSERT
  INTO "MYTABLE"
  VALUES('Manavallan','19BCS087','ASSISTANT LEADER','manavallan@gmail.com','M@n@');
INSERT
  INTO "MYTABLE"
  VALUES('Joodith','19BCS014','LEADER','joodith@gmail.com','J00d0');
INSERT
  INTO "MYTABLE"
  VALUES('Raghu','19BCS108','WORKER 1','raghu@gmail.com','r@gh^');
```

```
INSERT
 INTO "MYTABLE"
 VALUES('Praniis','19BCS060','WORKER 2','praniis@gmail.com','pr@n11s');
INSERT
 INTO "MYTABLE"
 VALUES('Dummy baba','19BCS000','WORKER 3','dummybaba@gmail.com','$^mm^');
delete from "MYTABLE" where Name='Dummy baba';
UPDATE "MYTABLE" SET EMAIL='Herminna@gmail.com' where name='Joodith';
```

3. Connect python code to db2.

```
conn = ibm_db.connect("DATABASE=bludb;"
            "HOSTNAME= 6667d8e9-9d4d-4ccb-ba32-
21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud "
            "PORT=30376;"
            "SECURITY=SSL;"
            "SSLServerCertificate=/home/joodo/myproject/DigiCertGlobalRootCA.cer;"
            "UID=bxc34720;"
            "PWD= ONWzqZktqrOwoocG;", "", "")
```

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

App.py

```
from flask import Flask, render_template, request, redirect, url_for, session
import ibm_db
import re
app = Flask(__name__)
app.secret_key = 'Zenik'
conn = ibm_db.connect("DATABASE=bludb;"
            "HOSTNAME=ba99a9e6-d59e-4883-8fc0-
d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;"
            "PORT=31321;"
            "SECURITY=SSL;"
```

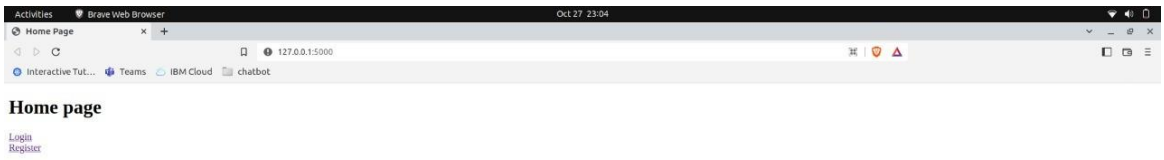
```
                "SSLServerCertificate=DigiCertGlobalRootCA.cer;"
                "UID=bxc34720;"
                "PWD=PIzsoX1FFpyquMfI;", "", "")

@app.route('/')
@app.route('/home')
def home():
    user={'auth':False}
    return render_template('landing.html', title='Home', msg=" ",user=user)



@app.route('/dashboard')
def dashboard():
    sql = "SELECT * FROM STUDENTS WHERE NAME=?"
    stmt = ibm_db.prepare(conn, sql)
    print(session['username'])
    ibm_db.bind_param(stmt, 1, session['username'])
    ibm_db.execute(stmt)
    account = ibm_db.fetch_assoc(stmt)
    user={'auth':True}
    return render_template('landing.html', title='Dashboard', account=account,user=user)



@app.route('/logout')
def logout():
    session.pop('Loggedin', None)
    session.pop('id', None)
    session.pop('username', None)
    return redirect('/')
```

```
@app.route('/success')
def success():
    return render_template('success.html')



@app.route('/login', methods=['GET', 'POST'])
def login():
    global userid
    msg = " "
    user = {'auth': False}
    if request.method == "POST":
        print("req post")
        username = request.form['username']
        password = request.form['password']
        sql = "SELECT * FROM STUDENTS WHERE NAME=? 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['NAME']
            userid = account['NAME']
            session['username'] = account['NAME']
            return redirect('/dashboard')
        else:
            msg = "Incorrect login credentials"
            return render_template('login.html', title='Login', msg=msg,user=user)
    else:
```

```
        return render_template('login.html', title='Login', msg=msg,user=user)



@app.route('/register', methods=['GET', 'POST'])
def register():
    user = {'auth': False}
    msg = " "
    print(request.method)
    if request.method == "POST":
        print("INside")
        username = request.form['username']
        print(username)
        email = request.form['email']
        password = request.form['password']
        password1 = request.form['re_password']
        rollno = request.form['rollno']
        sql = "SELECT * FROM STUDENTS WHERE NAME =? or EMAIL=?"
        print(username, email)
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt, 1, username)
        ibm_db.bind_param(stmt, 2, email)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
        print(account)
        if account:
            msg = "Account already exists"
        elif password1 != password:
            msg = "re-entered password doesnt match"
        elif not re.match(r'[A-Za-z0-9]+', username):
            msg = "Username should be only alphabets and numbers"
        else:
```

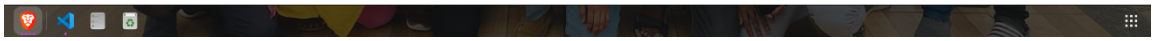
```
        print("insert")
        sql = "INSERT INTO STUDENTS VALUES (?,?,?,?,?)"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt, 1, username)
        ibm_db.bind_param(stmt, 2, rollno)
        ibm_db.bind_param(stmt, 3, "role")
        ibm_db.bind_param(stmt, 4, email)

        ibm_db.bind_param(stmt, 5, password)

        ibm_db.execute(stmt)
        return redirect('/login')
    return render_template('index.html', msg=msg, title="Register",user=user)


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

Dashboard:



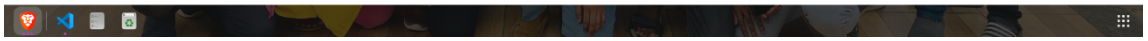
Register now!!

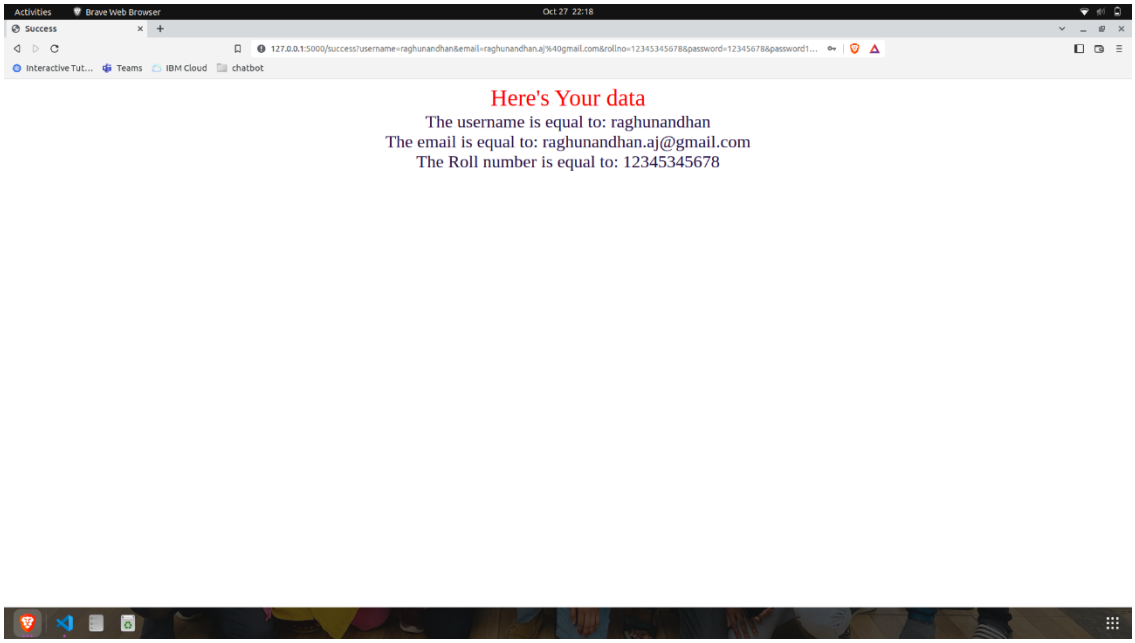
Username :
Email :
Roll number:
Password :
Password :
Submit



Register now!!

Username : raghunandhan
Email : raghunandhan.aj@gmail.com
Roll number: 12345345678
Password : ••••••••
Password : ••••••••
Submit




Activities
Brave Web Browser
Oct 27 22:18
Success
127.0.0.1:5000/success?username=raghunandhan&email=raghunandhan.aj%40gmail.com&rollno=12345345678&password=12345678&password1...
Interactive Tut...
Teams
IBM Cloud
chatbot
Here's Your data
The username is equal to: raghunandhan
The email is equal to: raghunandhan.aj@gmail.com
The Roll number is equal to: 12345345678