ASSIGNMENT-2

Student Name	ILAKKIYA .S
Roll number	510419106010
Team ID	PNT2022TMID29528

1.Create user table with user with email,username,password,roll no

2. perform update ,delete queries with user table

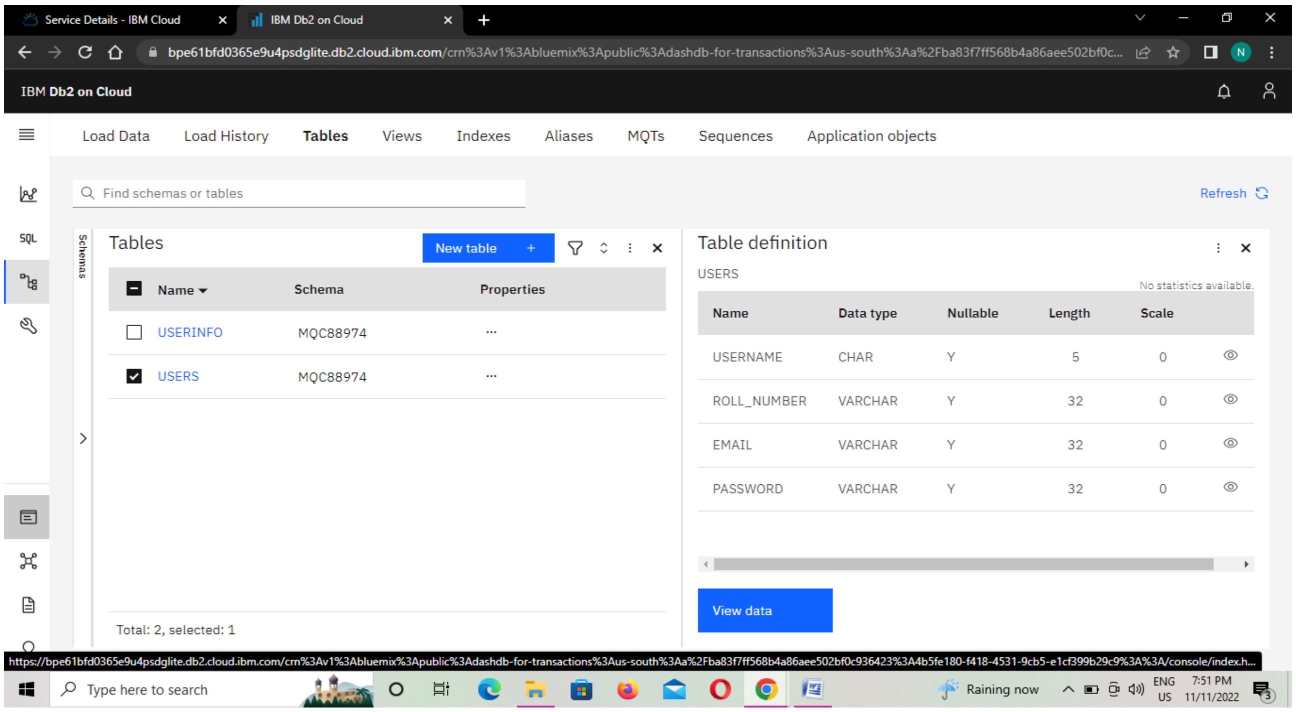
3.connect python code to database2

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 amnd password.if the user is valid show the welcome page

1.Create user table with user with email,username,password,roll no.

```
CREATE TABLE users(
        username char,roll_number varchar(32),email
varchar(32),password varchar(32))
```

ASSIGNMENT-2



Tables

Name	Schema	Properties
USERINFO	MQC88974	...
USERS	MQC88974	...

Total: 2, selected: 1

Table definition

USERS

No statistics available.

Name	Data type	Nullable	Length	Scale
USERNAME	CHAR	Y	5	0
ROLL_NUMBER	VARCHAR	Y	32	0
EMAIL	VARCHAR	Y	32	0
PASSWORD	VARCHAR	Y	32	0

https://bpe61bfd0365e9u4psdglite.db2.cloud.ibm.com/crn%3Av1%3Abluemix%3Apublic%3Adashdb-for-transactions%3Aus-south%3Aa%2Fba83f7ff568b4a86aee502bf0c936423%3A4b5fe180-f418-4531-9cb5-e1cf399b29c9%3A%3A/console/index.h...

2. perform update ,delete queries with user table

INSERT STATEMENT

insert into users values('niha',18,'gamma@gmail.com',72);

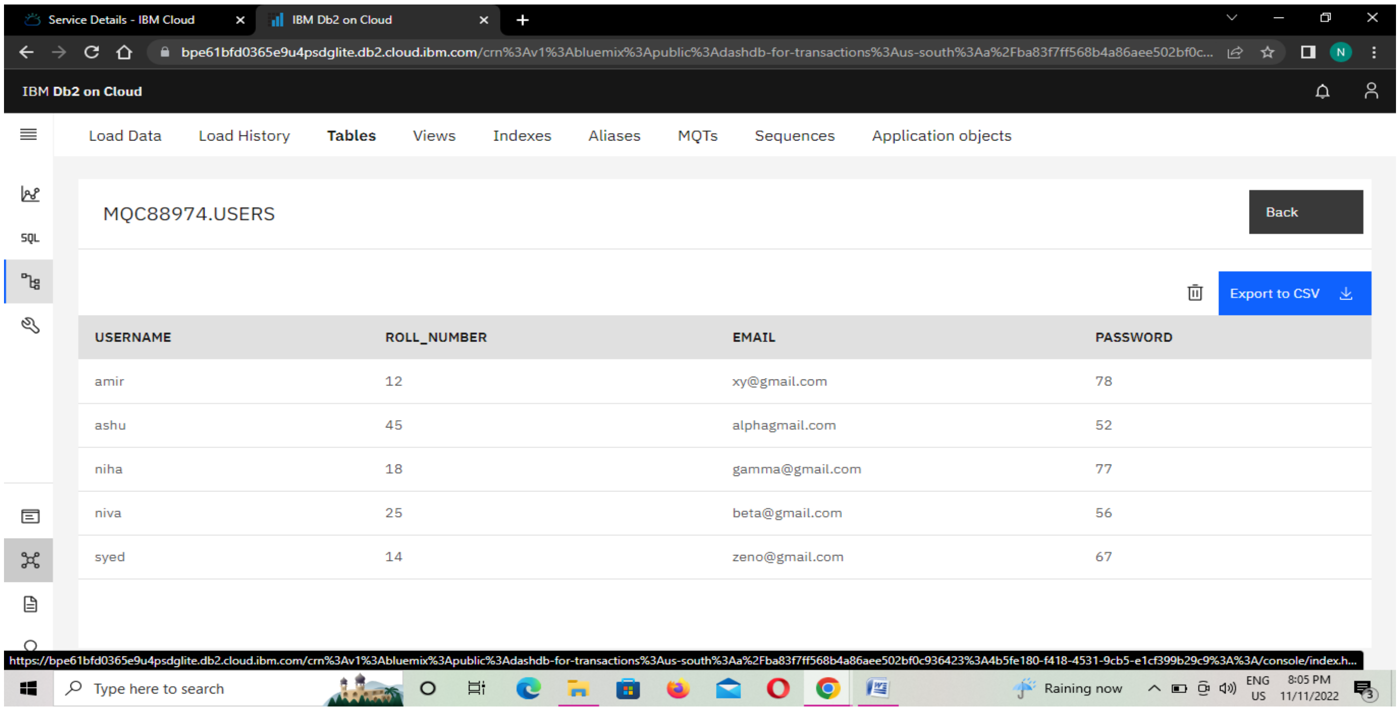
insert into users values('amir',12,'xy@gmail.com',78);

insert into users values('syed',14,'zeno@gmail.com',67);

insert into users values('niva',25, 'beta@gmail.com',56);

insert into users values('ashu',45,'alphagmail.com',52);

ASSIGNMENT-2

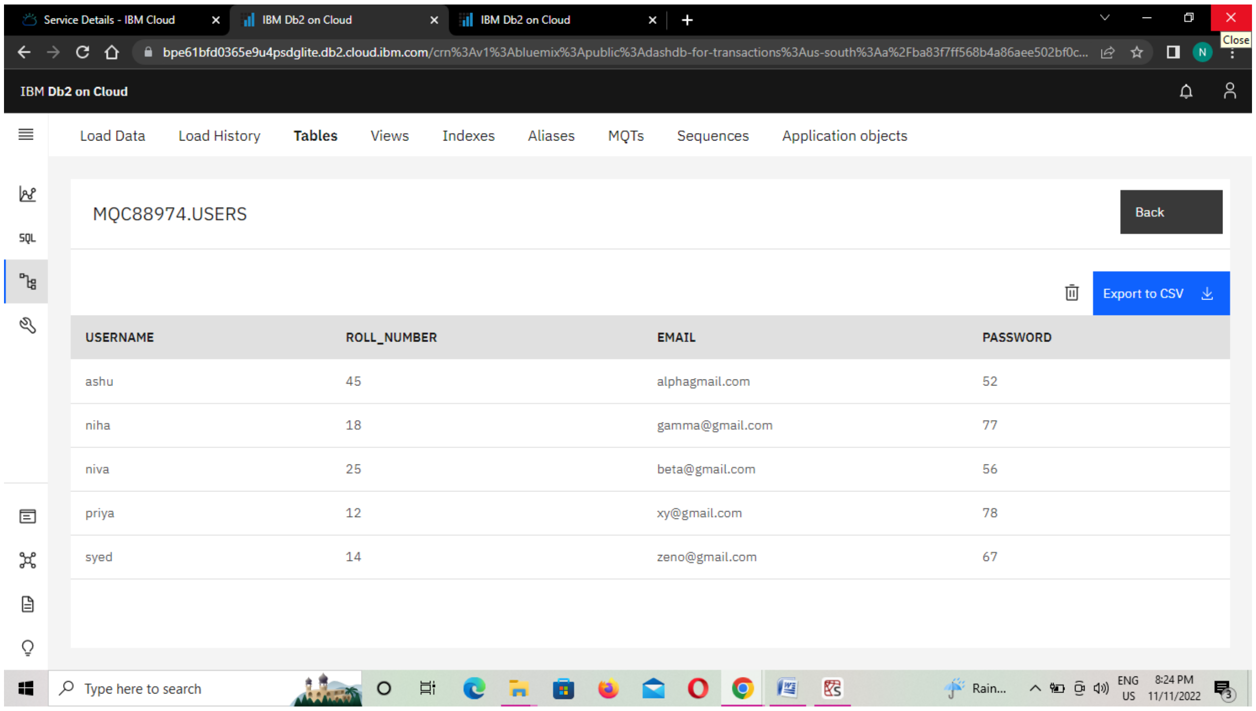


MQC88974.USERS

USERNAME	ROLL_NUMBER	EMAIL	PASSWORD
amir	12	xy@gmail.com	78
ashu	45	alphagmail.com	52
niha	18	gamma@gmail.com	77
niva	25	beta@gmail.com	56
syed	14	zeno@gmail.com	67

UPDATE STATEMENT

```
update users SET username='priya' WHERE
roll_number='12'
```



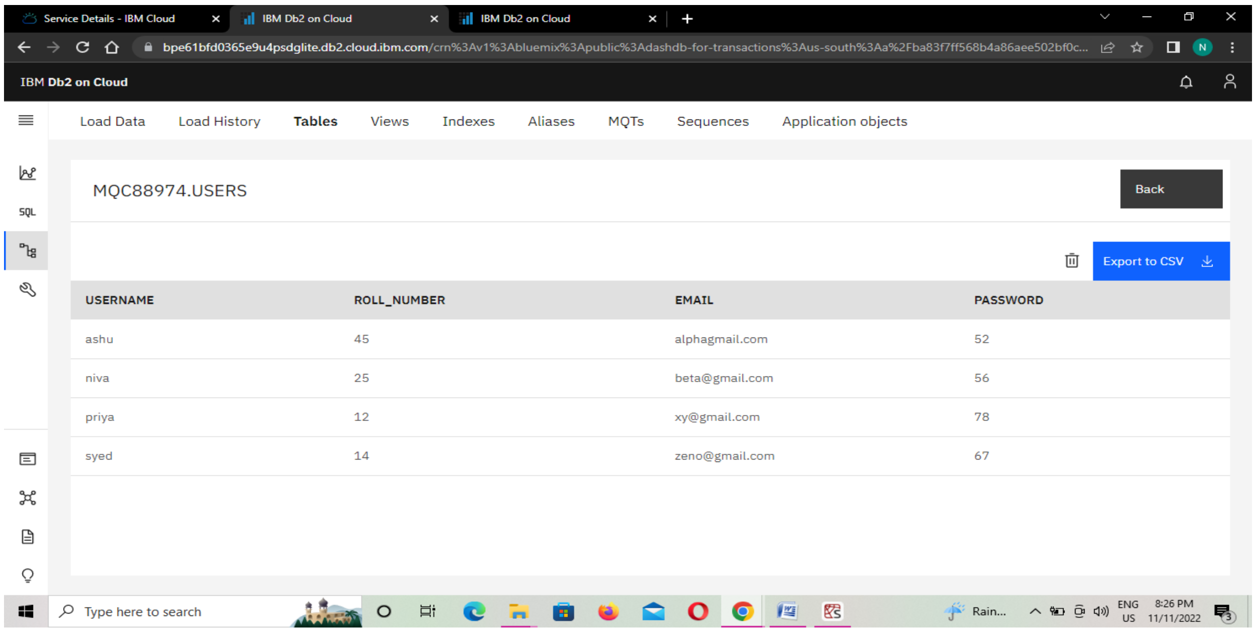
MQC88974.USERS

USERNAME	ROLL_NUMBER	EMAIL	PASSWORD
ashu	45	alphagmail.com	52
niha	18	gamma@gmail.com	77
niva	25	beta@gmail.com	56
priya	12	xy@gmail.com	78
syed	14	zeno@gmail.com	67

ASSIGNMENT-2

DELETE STATEMENT

delete from users WHERE roll_number='18'



MQC88974.USERS

USERNAME	ROLL_NUMBER	EMAIL	PASSWORD
ashu	45	alphagmail.com	52
niva	25	beta@gmail.com	56
priya	12	xy@gmail.com	78
syed	14	zeno@gmail.com	67

3.connect python code to database2

conn=ibm_db.connect("DATABASE=bludb;HOSTNAME=98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud";PORT=30875;security=;SSLServercertificateC:\Users\Dell\Desktop\job; UID=mqc88974";PWD=mbf14fY4F8mTNrXJ;)

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 amnd password.if the user is valid show

ASSIGNMENT-2

```
the welcome page 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=98538591-7217-
4024-b027-
8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud";P
ORT=30875;security=;SSLServercertificateC:\Users\Dell\Desktop\job
portal=;UID=mqc88974"

    ;PWD=mbf14fY4F8mTNrXJ;")


@app.route('/')

def home():

   return render_template('home.html')


@app.route("/login",methods=['GET',"POST"])

def login():
```

ASSIGNMENT-2

```
global userid
msg=" "

if request.method=="POST":
   username=request.form['username']
   password=request.form['password']
   sql="SELECT*FORM USER WHERE username=? AND password=?"
   stmt=ibm_db.prepare(conn,sql)
   ibm_db.bind_param(stmt,1,username)
   ibm_db.blind_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!'
```

ASSIGNMENT-2

```
            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 register():
    msg="  "
    if request.method=="POST":
        username=request.form['username']
        email=request.form['email']
        password=request.form["password"]
        sql="SELECT*FORM 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)
```

ASSIGNMENT-2

```
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 USER 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 sucessfully logged in!'

elif request.method=='post':
    msg='please fill out of the form'
    return render_template('register.html',msg=msg)
```

ASSIGNMENT-2

```
@app.route('/dashboard')

def dash():

    return render_template('dashboard.html')




@app.route('/apply',method==['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.fetc_assoc(stmt)
```

ASSIGNMENT-2

```
    print(account)
    if account:
        msg="There is only 1 job position!"
        return render_template('apply.html',msg=msg)

    insert_sql="INSERT INTO JOB VALUES(?,?,?,?,?)"
    prep_stmt=ibm_db.prepare(conn,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 the job position"
    session['loggedin']=True

elif request.method=="POST":
    msg='please fill out the form'
    return render_template()
```

ASSIGNMENT-2

```
elif request.method=='POST':
    msg='please fill'
```