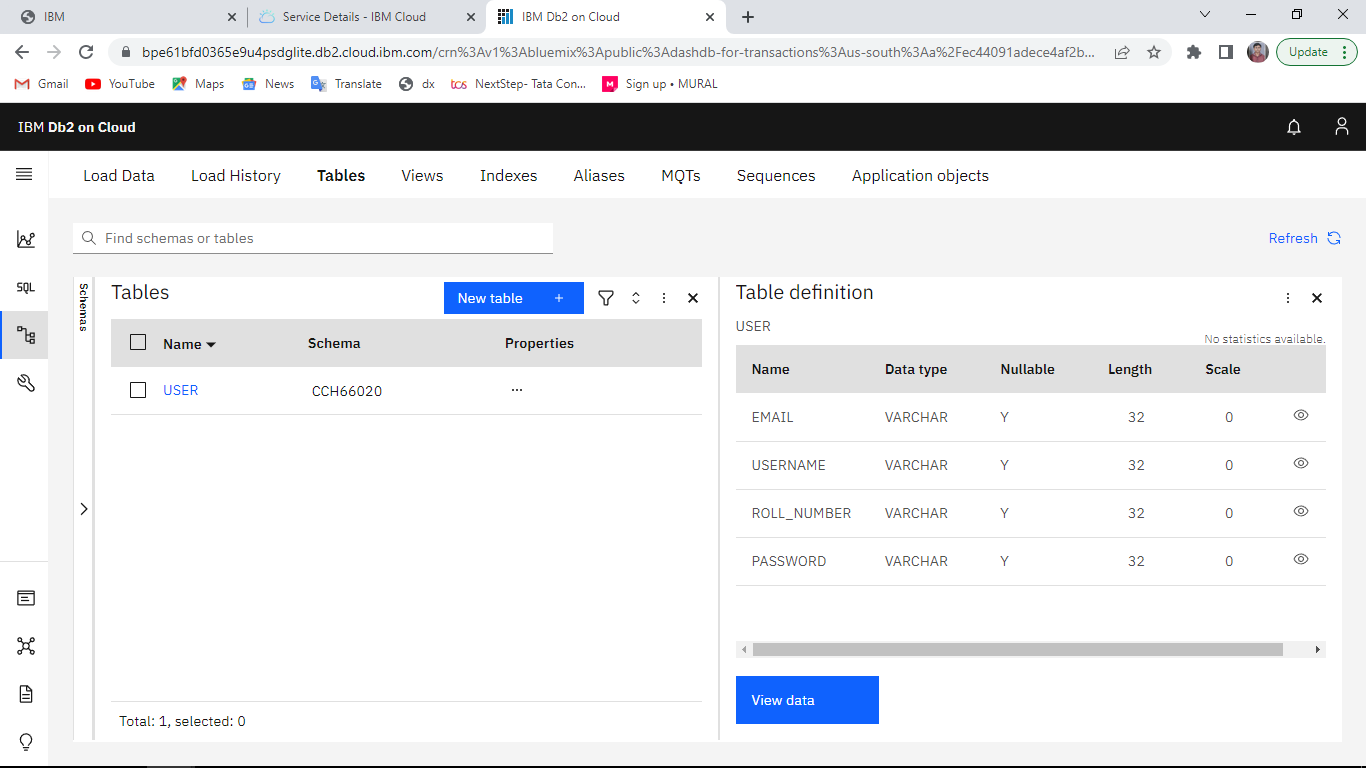
# Assignment -2

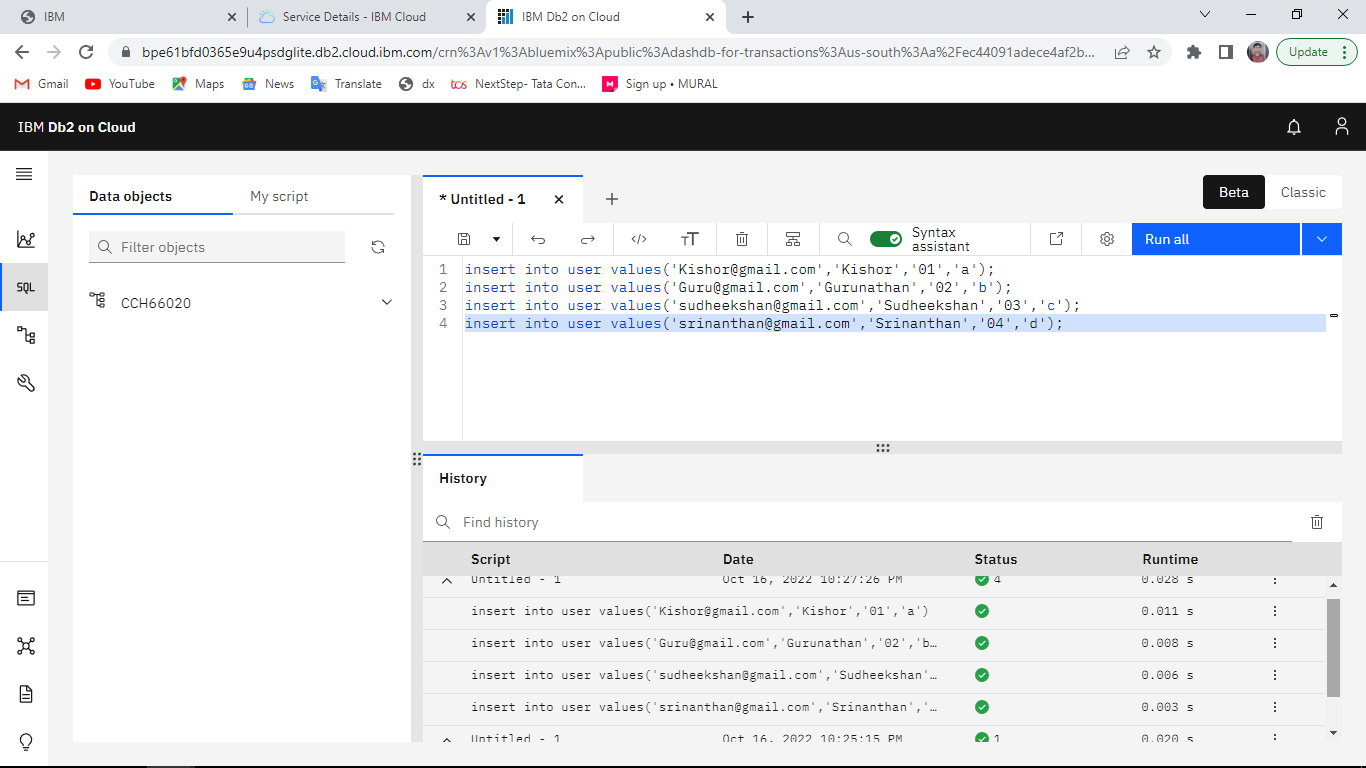
|  |  |
| --- | --- |
| Assignment Date | 26 September 2022 |
| Student Name | Gurunathan P |
| Student Roll Number | 311519104030 |
| Maximum Marks | 2 Marks |

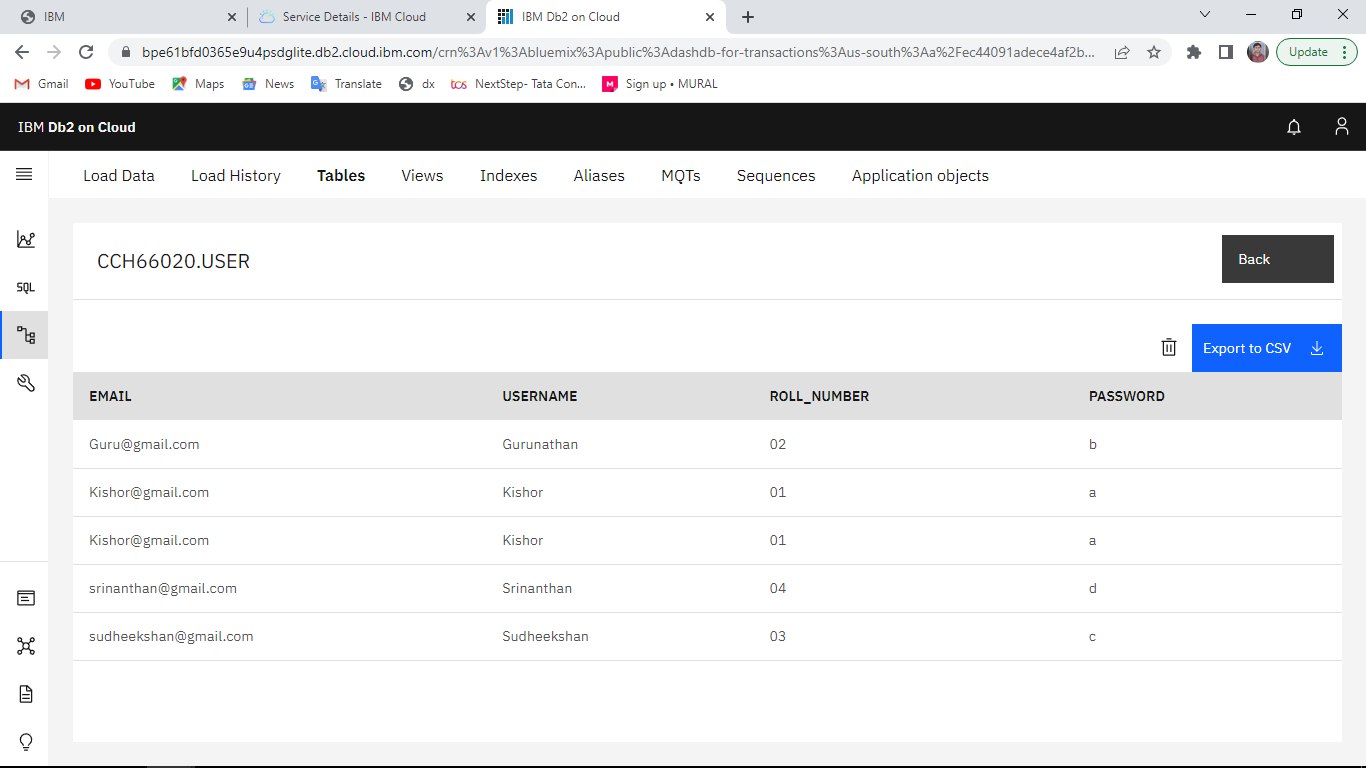
**Question-1:**

Create User table with email, username, roll number, password.



* insert into user values('Kishor@gmail.com','Kishor','01','a');
* insert into user values('Guru@gmail.com','Gurunathan','02','b');
* insert into user values('Sudheekshan@gmail.com','Sudheekshan','03','c');
* insert into user values('Srinanthan@gmail.com','Srinanthan','04','d');

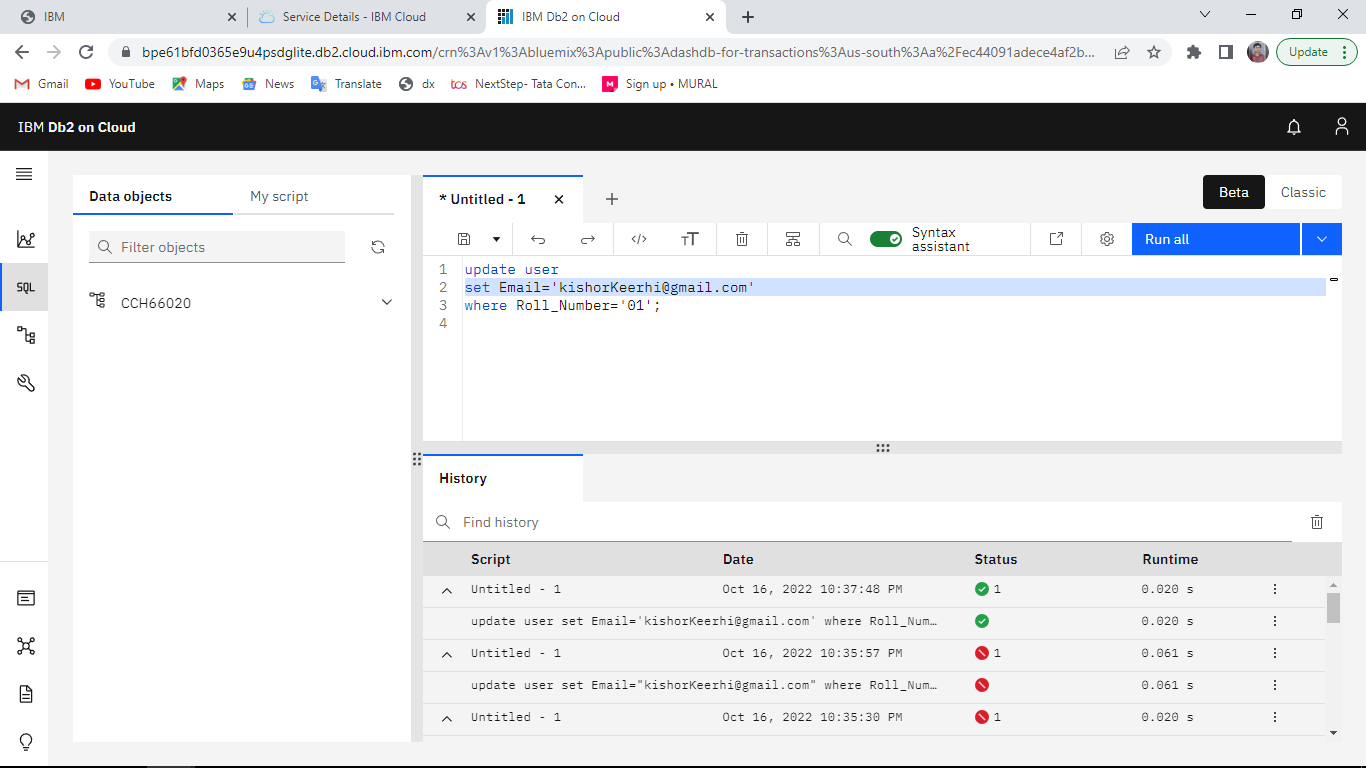


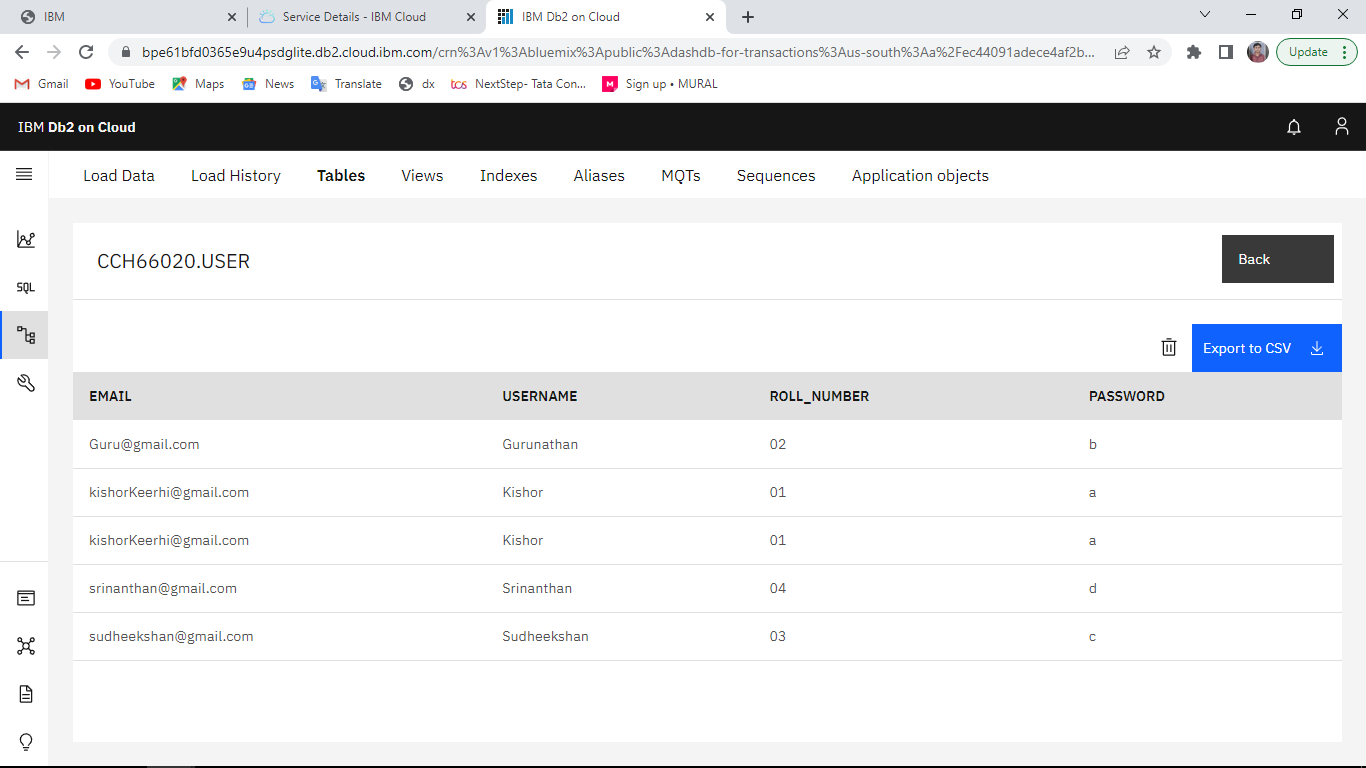


# Question-2:

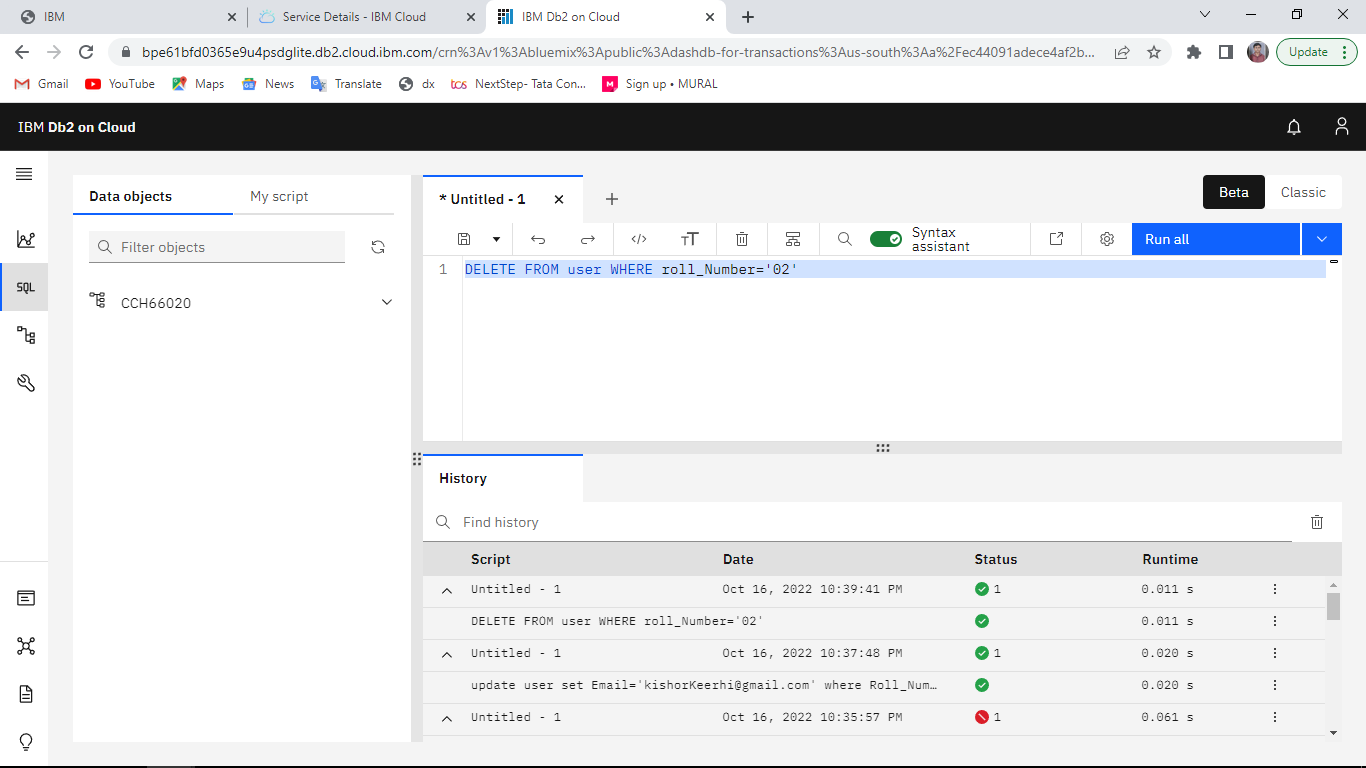
Perform UPDATE, DELETE Queries with user table

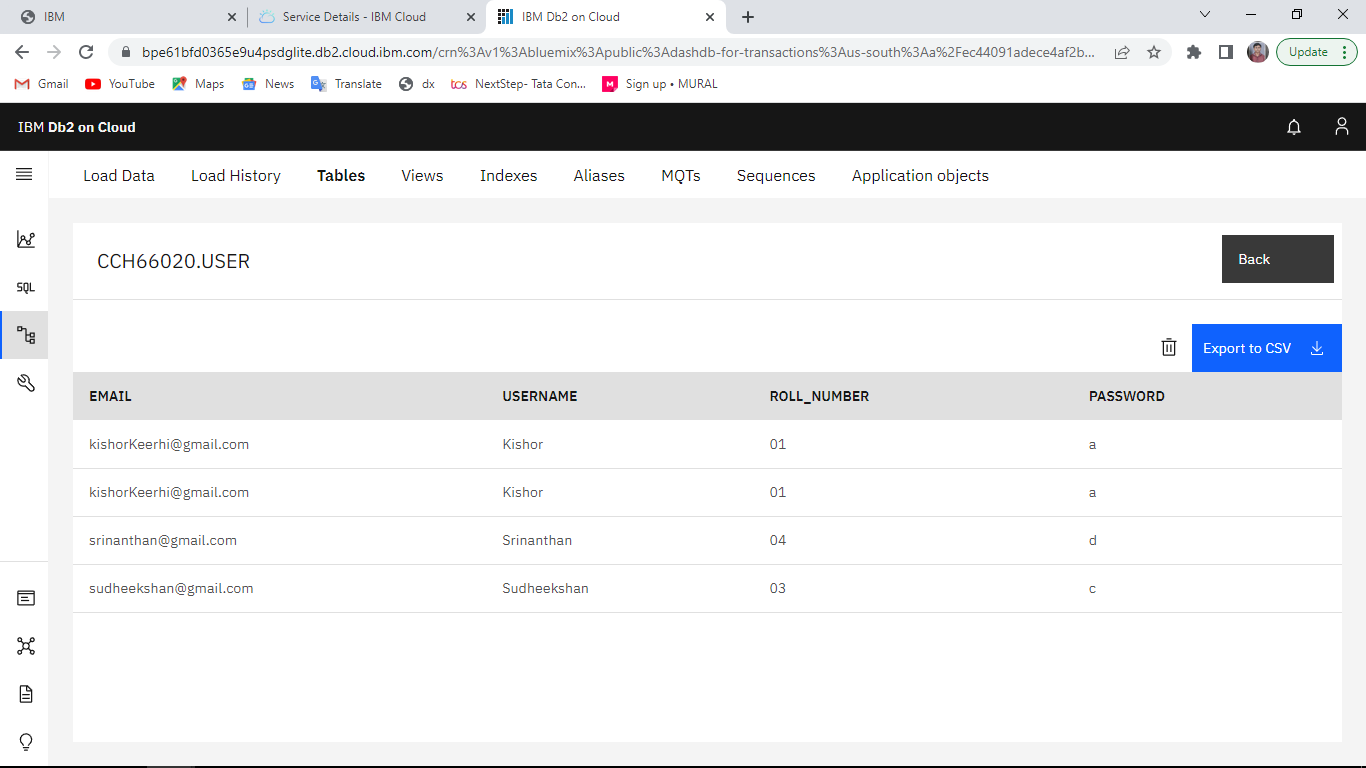
1.update user set email='KishorKeerthi@gmail.com' where roll\_number='01';





**DELETE FROM user WHERE roll\_number=’03’;**





# Question 3:

Connect python code to db2

# Program:

from flask import Flask import ibm\_db

app=Flask( name ) try:

conn=ibm\_db.connect("DATABASE=bludb;HOSTNAME=2f3279a5-73d1-4859-88f0-a6c3e6b4b907.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud;PORT=30756;SECURITY=SSL;SSLSe rverCertificate=DigiCertGlobalRootCA.cr;UID=cch66020;PWD=gxVsoGjnqdx8sTAh","","")

except:

print("Unable to connect: ",ibm\_db.conn\_error())

@app.route("/") def dashboard():

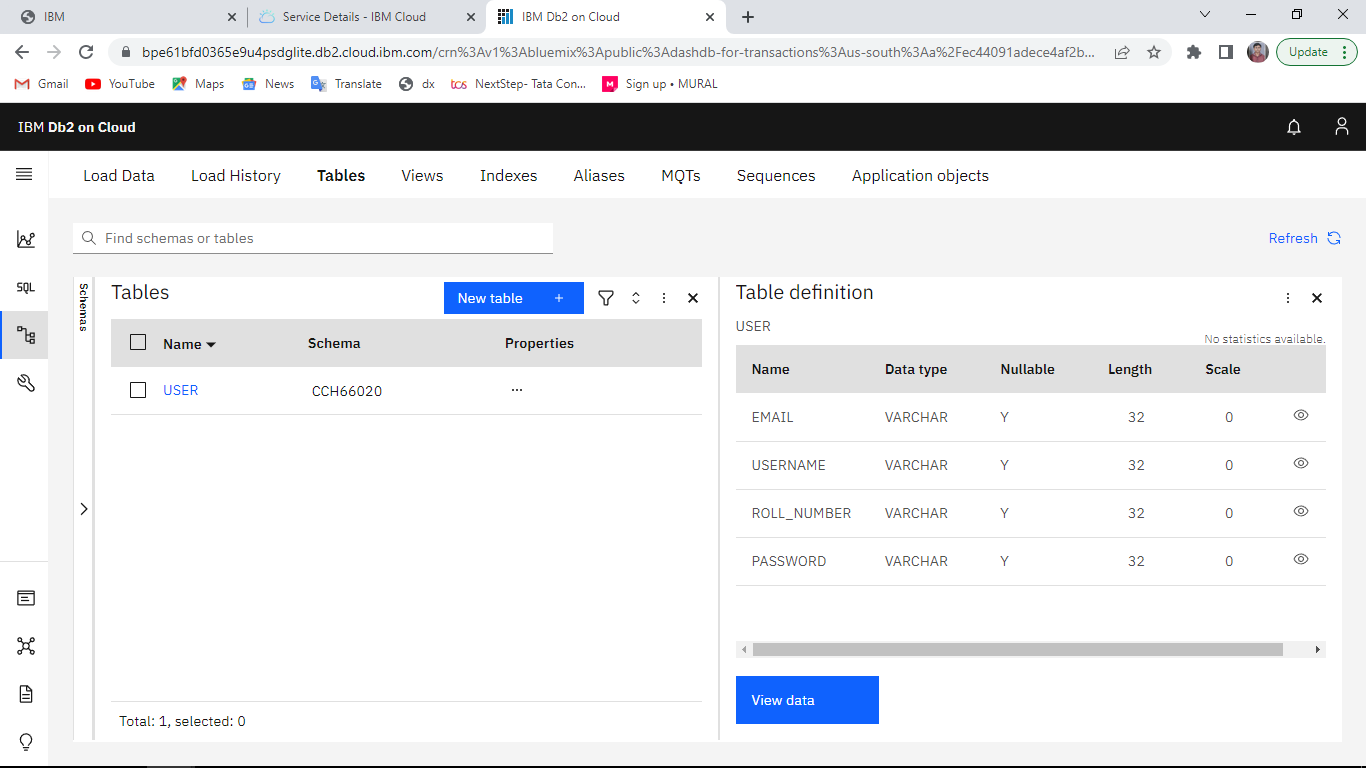
return “welcome”

if name ==' main ': app.run(debug=True)

# Question 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 username and password. If the user is valid show the welcome page.

# Database:



**Program:**

# User.py

from flask import Flask, render\_template,request,redirect,url\_for,session,flash

import ibm\_db

app=Flask( name

app.secret\_key='a'

try:

conn=ibm\_db.connect("DATABASE=bludb;HOSTNAME=6667d8e9-9d4d-4ccb-ba32- 21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=30376;SECURITY=SSL;SSLSe rverCertificate=DigiCertGlobalRootCA.crt;UID=prj41491;PWD=8NNe2bnsXzveXAcg","","")

except:

print("Unable to connect: ",ibm\_db.conn\_error())

@app.route("/") def dash():

return render\_template('register.html',msg=" ")

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

error = None

if request.method=='POST': username=request.form['username'] email=request.form['email'] roll\_number=request.form['rollnumber'] password=request.form['password']

sql="SELECT \* FROM user WHERE roll\_number=?" prep\_stmt=ibm\_db.prepare(conn,sql) ibm\_db.bind\_param(prep\_stmt,1,roll\_number) ibm\_db.execute(prep\_stmt) account=ibm\_db.fetch\_assoc(prep\_stmt) print(account)

if account:

error="Account already exists! Log in to continue !" else:

insert\_sql="INSERT INTO user 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,roll\_number) ibm\_db.bind\_param(prep\_stmt,4,password) ibm\_db.execute(prep\_stmt)

flash(" Registration successfull. Log in to continue !")

else:

pass

return render\_template('login.html',error=error)

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

error = None

if request.method=='POST': username=request.form['username'] password=request.form['password']

sql="SELECT \* FROM user 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'] session["username"]=account["USERNAME"] flash("Logged in successfully!")

return redirect(url\_for("welcome\_page")) else:

error="Incorrect username / password"

return render\_template('login.html',error=error)

@app.route('/welcome') def welcome\_page():

return render\_template("welcome.html",user=session['id'])

if name ==' main ': app.run(debug=True)

# register.html

<html>

<head>

<title>Registration page</title>

<link rel="stylesheet" href="{{url\_for('static', filename='style.css')}}">

</head>

<body>

<div class="center"> <br /><br />

<form action="/register" method="POST">

<h2>Student Registration Portal</h2> <br /><br />

<p> Enter Email ID:</p>

<input type="email" name="email"/>

<p> Enter Username:</p>

<input type="text" name= "username" />

<p> Enter Password:</p>

<input type="password" name="password"/>

<p> Enter Roll number:</p>

<input type="text" name="rollnumber"/>

</br></br></br>

<input type="submit" value="submit" />

</form>

</div>

</body>

</html>

# login.html

<html>

<head>

<title>Login page</title>

<link rel="stylesheet" href="{{url\_for('static', filename='style.css')}}">

</head>

<body>

{% if error %}

<p><strong style="color:red">Error</strong>: {{error}}</p>

{% endif %}

{% with messages = get\_flashed\_messages() %}

{% if messages %}

{% for message in messages %}

<p style="color:green">{{ message }}</p>

{% endfor %}

{% endif %}

{% endwith %}

<div class="center"> <br /><br />

<form action="/login" method="POST">

<h1>Login</h1> <br /><br />

<p> Enter Username:</p>

<input type="text" name= "username" />

<p> Enter Password:</p>

<input type="password" name="password"/>

</br></br></br>

<input type="submit" value="submit" />

</form>

</div>

</body>

</html>

# welcome.html

<html>

<head>

<title>Login page</title>

<link rel="stylesheet" href="{{url\_for('static', filename='style.css')}}">

</head>

<body>

{% with messages = get\_flashed\_messages() %}

{% if messages %}

{% for message in messages %}

<p style="color:green">{{ message }}</p>

{% endfor %}

{% endif %}

{% endwith %}

<br /><br />

<center>

<h2 > User {{user}}</h2> <br />

<img src="{{url\_for('static', filename='welcome.jpeg')}}" /><br /><br />

<p><i>As long as you keep yourself focused, you will have a mighty mind that can make everything possible.</i></h3>

</center><br/>

<div style="margin:40px">

<center><h1>Welcome</h1> </center>

</div>

</body>

</html>

# style.css

.center{ margin: auto; width: 15%;

}

h2{

text-align:center;

}

input{

border-radius:5px; padding:7px;

}

# Output:

