Assignment-2

Project Domain	Cloud Application Development
Project Title	Customer Care Registry.
Team ID	PNT2022TMID44404
Name	GOWTHAM R
Roll No	731119205009
Date	03rd Oct 2022

Questions:

1. Create registration page in html with username, email, and phone number and by using POST method display it in next html page.
2. Develop a flask program which should contain at least 5 packages used from pypi.org.
3. Create User table with user with email, username, roll number, password. 4. Perform UPDATE, DELETE Queries with user table
5. Connect python code to db2.
6. 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.

Answers:

1.Create registration page in html with username, email, and phone number and by using POST method display it in next html page.

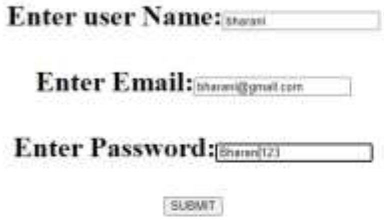
Login.html:

```
<html>
<body>
        <center>
        <form action = "http://localhost:3890/login" method = "post">
                <h1>
                Enter user Name:<input type = "text" name = "userName"/><br><br>
                Enter Email-id:<input type = "text" name = "emailId"/><br><br>
               Enter   Phone          Number:<input          type     =        "text"   name
 "phoneNumber"/><br><br>
                <input type = "submit" value = "SUBMIT"/>
```

```
            </h1>
            </form>                                                                                      =
            </center>
</body>
</html>
```

Sample.py:

```
from flask import Flask, redirect, url_for, request app
= Flask(__name__)

@app.route('/login', methods=['POST'])
def login():   if request.method == 'POST':
                    user_name = request.form['userName']                    email_id =
request.form['emailId']                     phone_number = request.form['phoneNumber']
          return  '{}{}{}{}{}{}'.format("<center><h1>Your  user      name  is:
",user_name,"</h1><br><br><h2>Your  email-id  is:  ",email_id,"</h2><br><br><h3>Your
phone number is: ",phone_number,"</h3></center>")

if __name__ == '__main__':
            app.run('127.0.0.1',3890)
```



Enter user Name: bharani

Enter Email: bharani@gmail.com

Enter Password: Bharani123

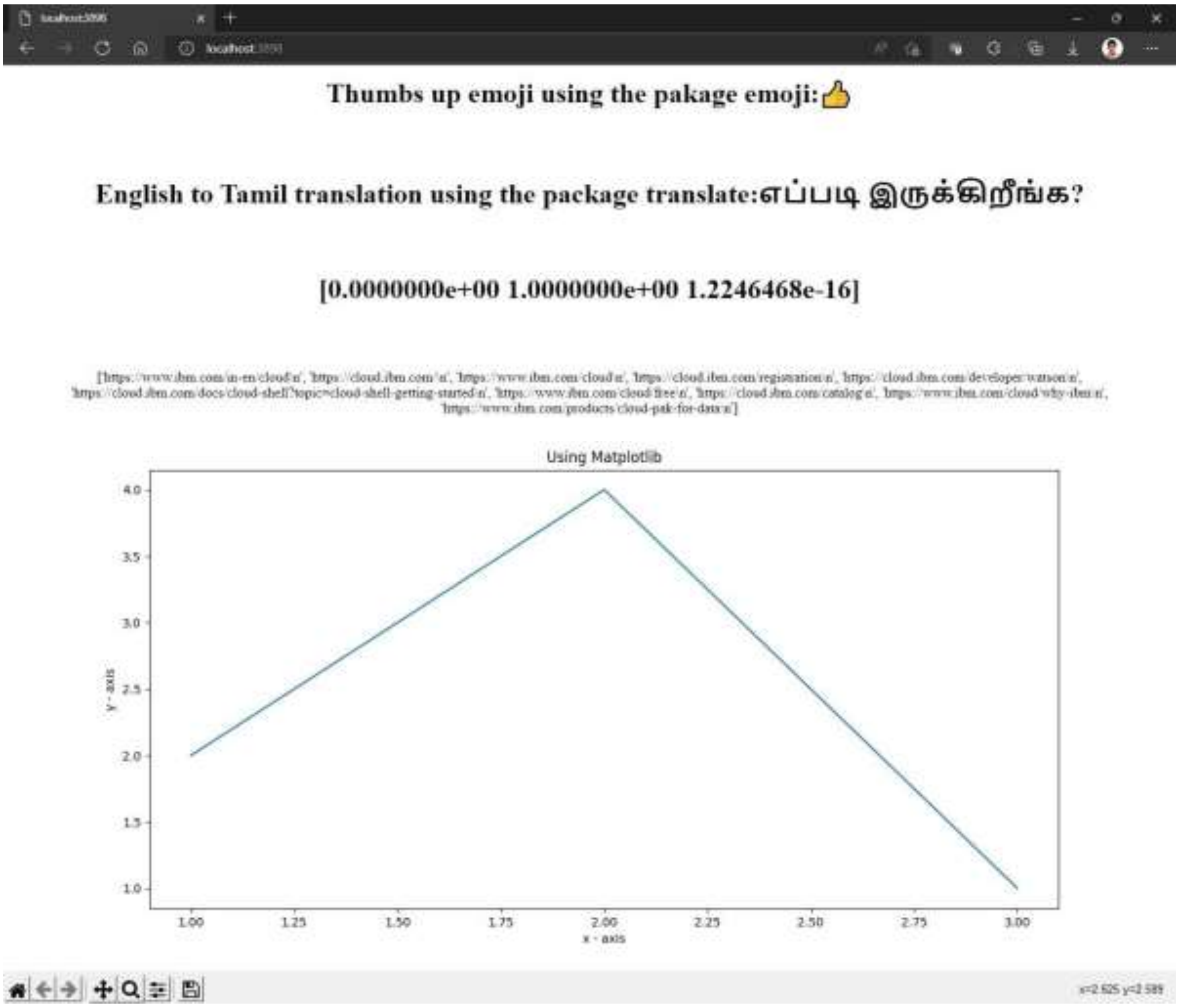
2.Develop a flask program which should contain at least 5 packages used from pypi.org.

Packages used:

Flask, emoji, matplotlib, numpy, translate, googlesearch

Packages.py:

```
from flask
import Flask from emoji
import emojize import
matplotlib.pyplot as plt
import numpy as np from
translate import Translator
from googlesearch import search

app=Flask(__name__)

@app.route('/') def login():     emojione=("Thumbs up emoji using the pakage
emoji:"+emojize(":thumbs_up:"))

    x = [1,2,3]      y = [2,4,1]
plt.plot(x, y)      plt.xlabel('x -
axis')      plt.ylabel('y - axis')
plt.title('Using Matplotlib')
plt.show()      a = np.array([0,
np.pi/2, np.pi])

    translator= Translator(to_lang="ta")
    translation = ("English to Tamil translation using the package
translate:"+translator.translate("How are you?"))

    query = "IBM Cloud"     tmp=search(query, tld="co.in",
num=10, stop=10, pause=2)     res=[]     for i in tmp:
res.append(i+"\n")

    return
("<center>"+"<h1>"+emojione+"</br></br></br>"+translation+"</br></br></br>"+str(np.
sin(a))+"</h1></br></br></br>"+str(res)+"</center>")

if __name__ == '__main__':
            app.run('127.0.0.1',3898,debug=True)
```



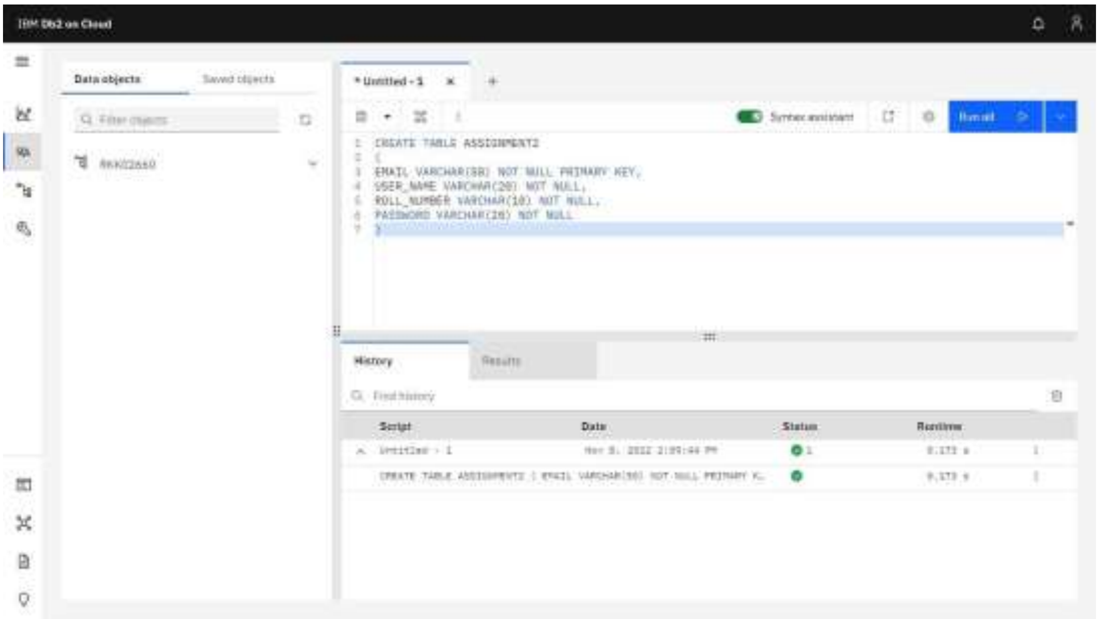
Thumbs up emoji using the pakage emoji:👍

English to Tamil translation using the package translate:எப்படி இருக்கிறீங்க?

[0.0000000e+00 1.0000000e+00 1.2246468e-16]

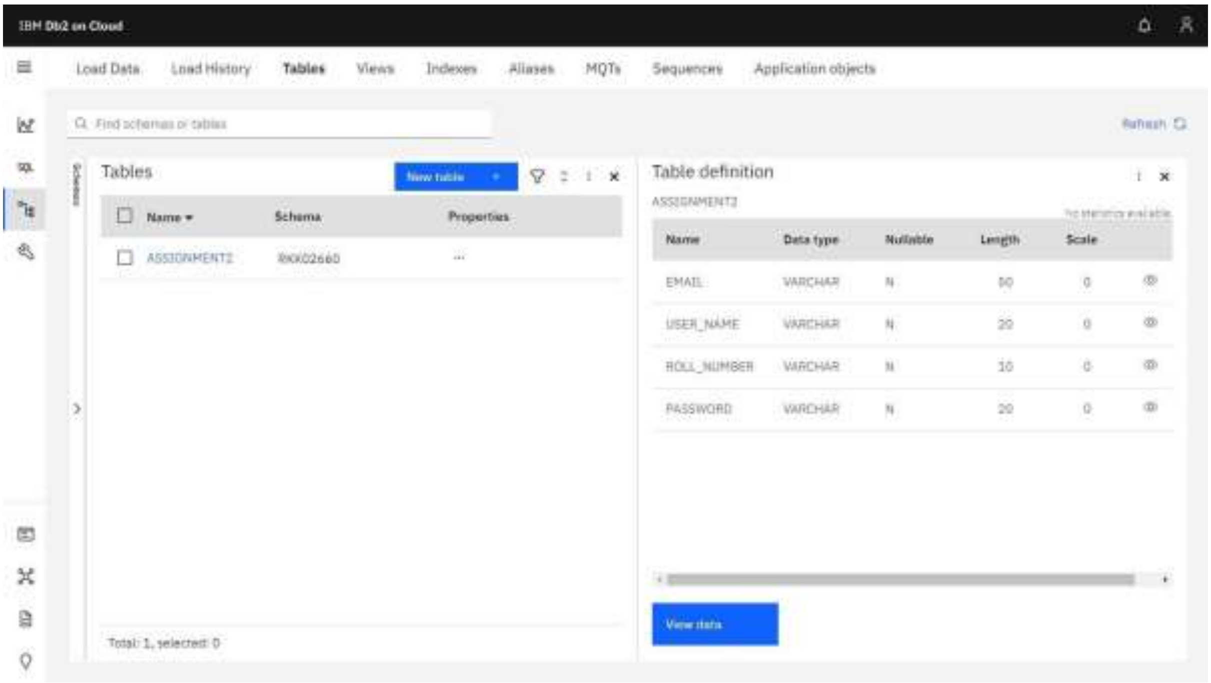
['https://www.ibm.com/in-en/cloud\n', 'https://cloud.ibm.com/\n', 'https://www.ibm.com/cloud\n', 'https://cloud.ibm.com/registration\n', 'https://cloud.ibm.com/developer/watson\n', 'https://cloud.ibm.com/docs/cloud-shell?topic=cloud-shell-getting-started\n', 'https://www.ibm.com/cloud/free\n', 'https://cloud.ibm.com/catalog\n', 'https://www.ibm.com/cloud/why-ibm\n', 'https://www.ibm.com/products/cloud-pak-for-data\n']

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



```
CREATE TABLE ASSIGNMENT2
(
EMAIL VARCHAR(50) NOT NULL PRIMARY KEY,
USER_NAME VARCHAR(20) NOT NULL,
ROLL_NUMBER VARCHAR(10) NOT NULL,
PASSWORD VARCHAR(20) NOT NULL
)
```

Script	Date	Status	Runtime
Untitled - 1	Nov 5, 2022 2:59:44 PM	● 1	0.173 s
CREATE TABLE ASSIGNMENT2 (EMAIL VARCHAR(50) NOT NULL PRIMARY K...		●	0.173 s

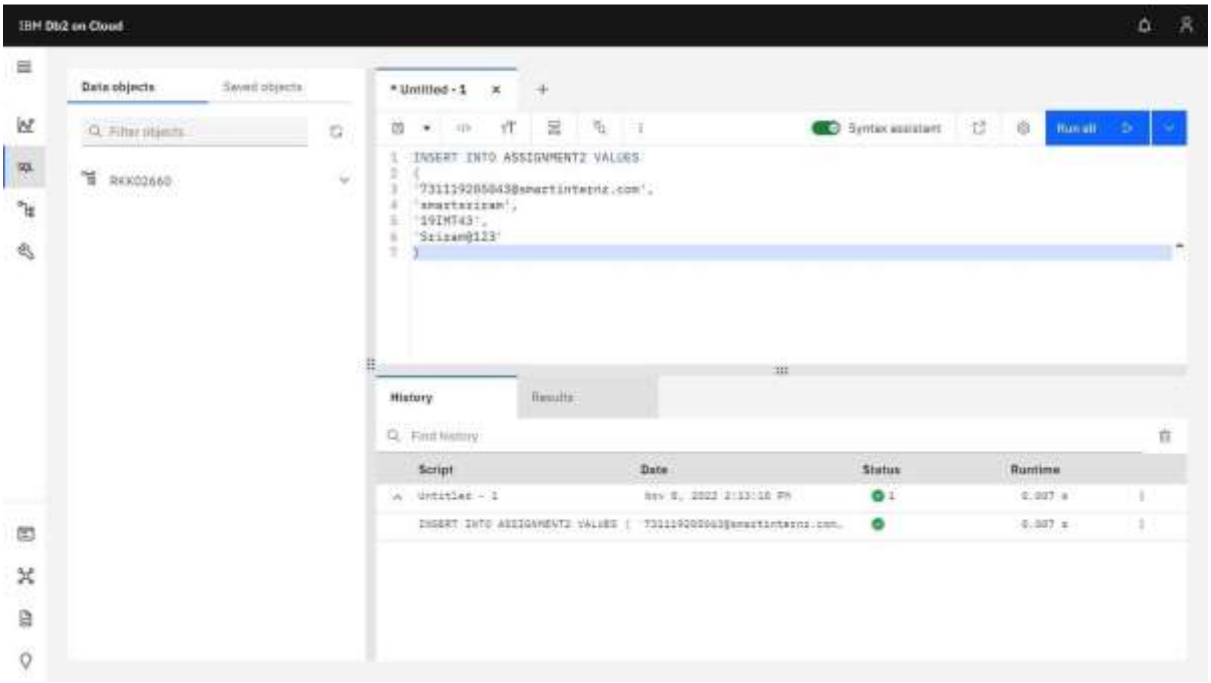


□	Name	Schema	Properties
□	ASSIGNMENT2	RKK02660	...

Total: 1, selected: 0

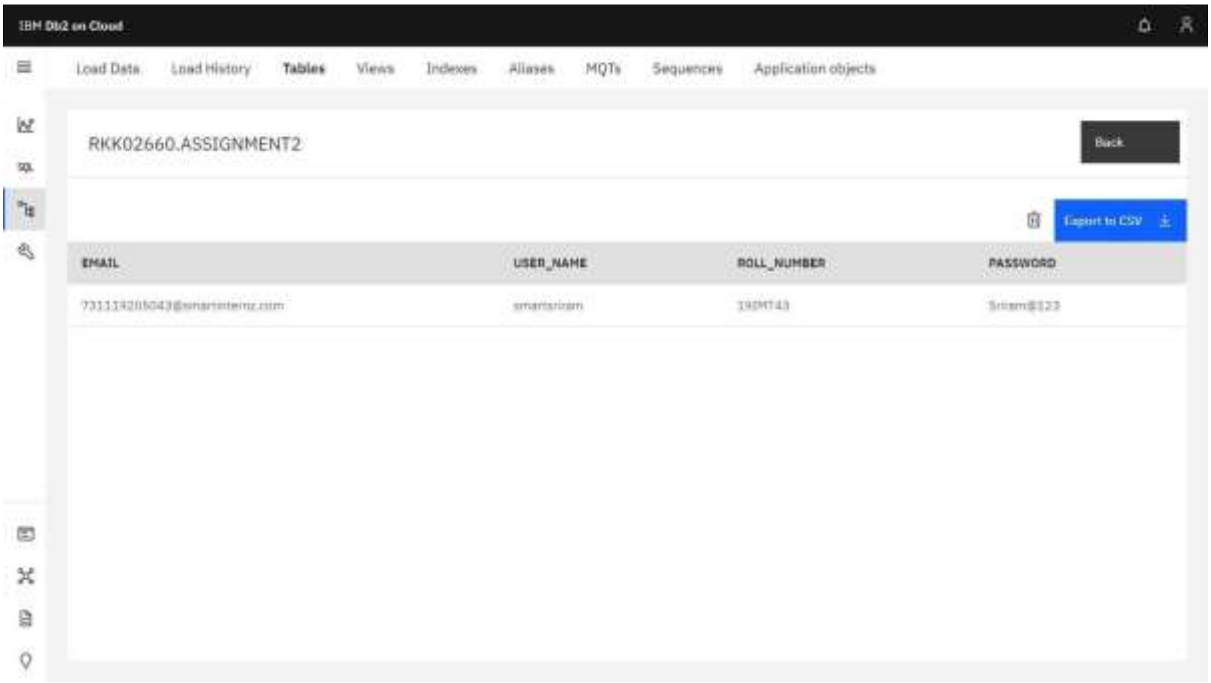
ASSIGNMENT2

Name	Data type	Nullable	Length	Scale
EMAIL	VARCHAR	N	50	0
USER_NAME	VARCHAR	N	20	0
ROLL_NUMBER	VARCHAR	N	10	0
PASSWORD	VARCHAR	N	20	0



```
INSERT INTO ASSIGNMENT2 VALUES
(
'731119205043@smartinternz.com',
'smartsriram',
'19IMT43',
'Sriram@123'
)
```

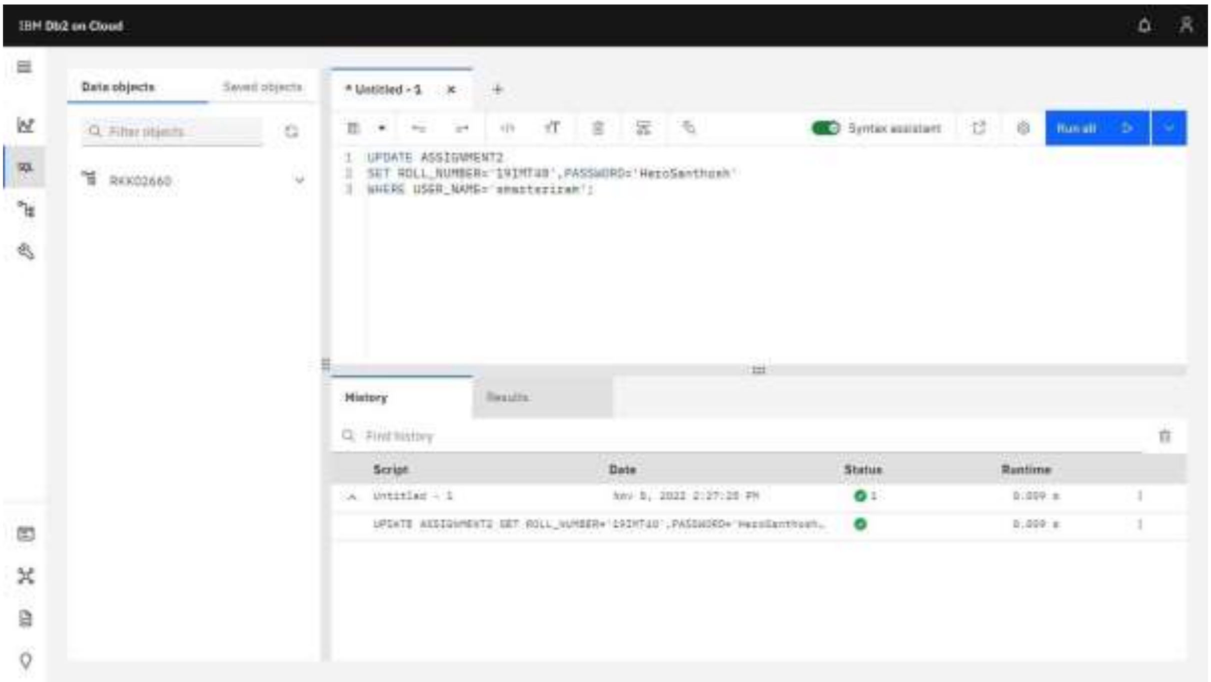
Script	Date	Status	Runtime
Untitled - 1	Nov 8, 2022 2:13:10 PM	● 1	0.007 s
INSERT INTO ASSIGNMENT2 VALUES (	731119205043@smartinternz.com...	●	0.007 s



RKK02660.ASSIGNMENT2

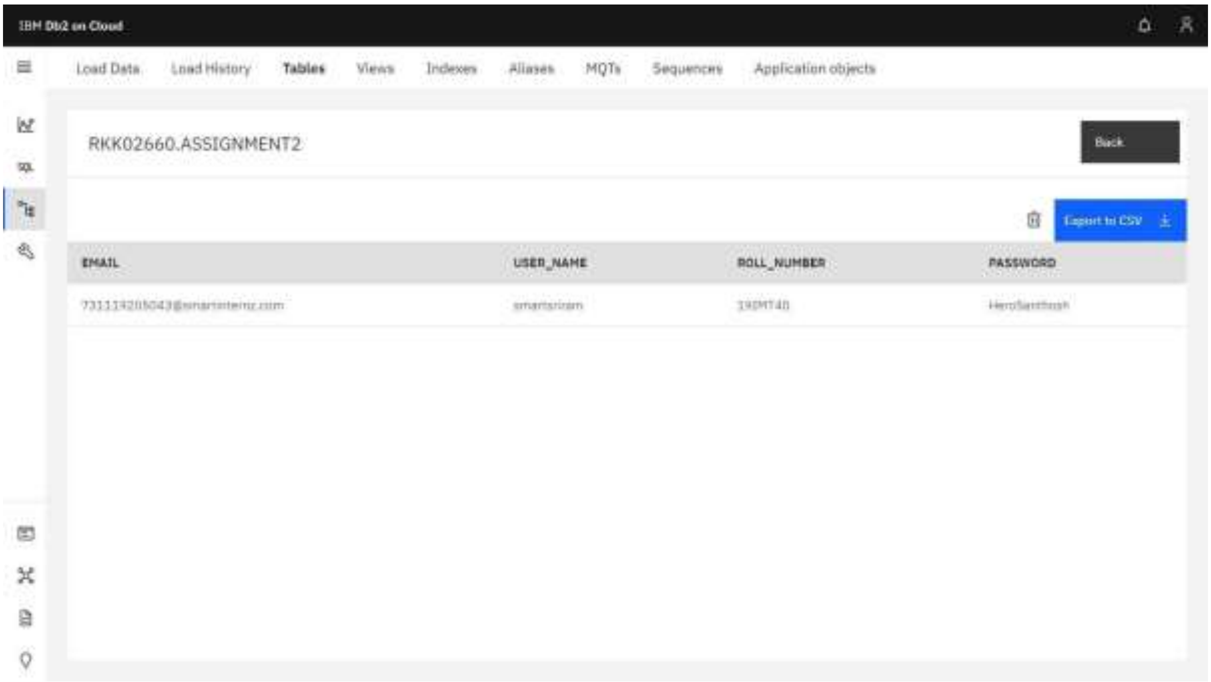
EMAIL	USER_NAME	ROLL_NUMBER	PASSWORD
731119205043@smartinternz.com	smartsriram	19IMT43	Sriram@123

-- 4.Perform UPDATE, DELETE Queries with user table



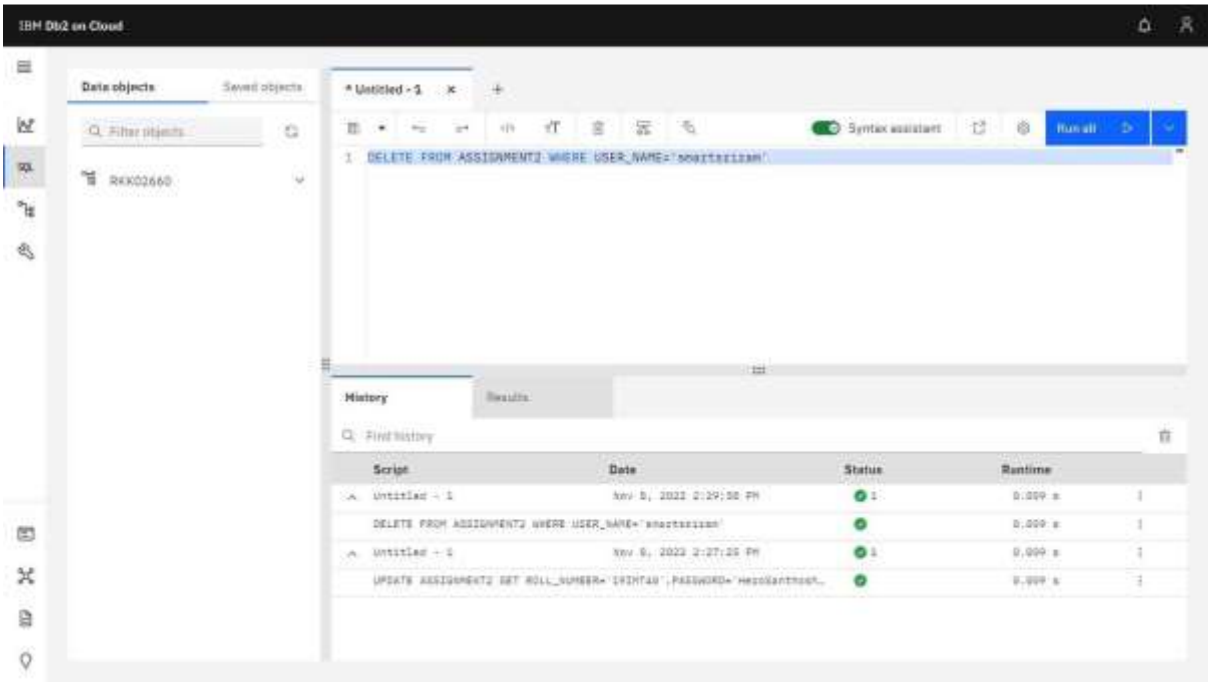
```
UPDATE ASSIGNMENT2
SET ROLL_NUMBER='19IMT40',PASSWORD='HeroSanthosh'
WHERE USER_NAME='smartsriram';
```

Script	Date	Status	Runtime
Untitled - 1	Nov 5, 2022 2:27:25 PM	✓ 1	0.009 s
UPDATE ASSIGNMENT2 SET ROLL_NUMBER='19IMT40',PASSWORD='HeroSanthosh...		✓	0.009 s



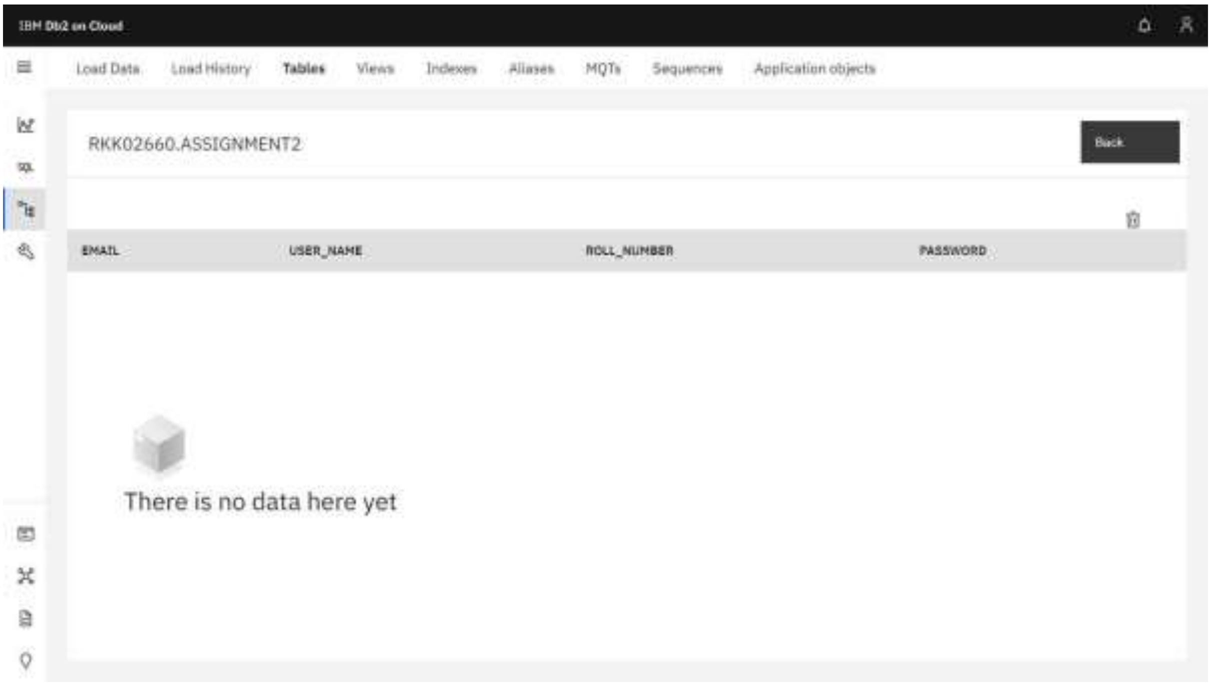
RKK02660.ASSIGNMENT2

EMAIL	USER_NAME	ROLL_NUMBER	PASSWORD
731119205043@smartinternz.com	smartsriram	19IMT40	HeroSanthosh



```
1  DELETE FROM ASSIGNMENT2 WHERE USER_NAME='smartsrizam'
```

	Script	Date	Status	Runtime
^	Untitled - 1	Nov 5, 2022 2:29:58 PM	✓ 1	0.009 s
	DELETE FROM ASSIGNMENT2 WHERE USER_NAME='smartsrizam'		✓	0.009 s
^	Untitled - 1	Nov 6, 2022 2:27:25 PM	✓ 1	0.009 s
	UPDATE ASSIGNMENT2 SET ROLL_NUMBER='191MT40',PASSWORD='HeroSanthosh…		✓	0.009 s



RKK02660.ASSIGNMENT2

EMAIL	USER_NAME	ROLL_NUMBER	PASSWORD

There is no data here yet

----

5. Connect python code to db2.

6. 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

login.html: <html> <body>

```
        <center>
```

```
                    <form action = "http://localhost:3899/login" method = "post">
                                        <h1>
                                        Enter user Name:<input type = "text" name =
"username" required /><br><br>
                                        Enter Password:<input type = "text" name =
"password"/><br><br>
    <input type = "submit" value = "SUBMIT" name="submit"/><br><br>
                                        <a href="/regis">Click here to register</a>
                              </h1>
                    </form>
          </center>
</body>
```

```
</html>
```

register.html:

```
<html>
<body>
                                <center>
                                                                                    <form action
=
"http://localhost:3899/register" method = "post">
                                                  <h1>
                                          Enter user Name:<input type = "text" name =
"username" required /><br><br>
                                                                                             Enter
Email:<input type =
"text" name =
"email"/><br><br>
                                                  Enter Password:<input type = "text"
name =  "password"/><br><br>
                                          <input type = "submit" value = "SUBMIT"/>
                                        </h1>
                                          </form>
                                </center>
</body>
</html>
```

welcome.html:

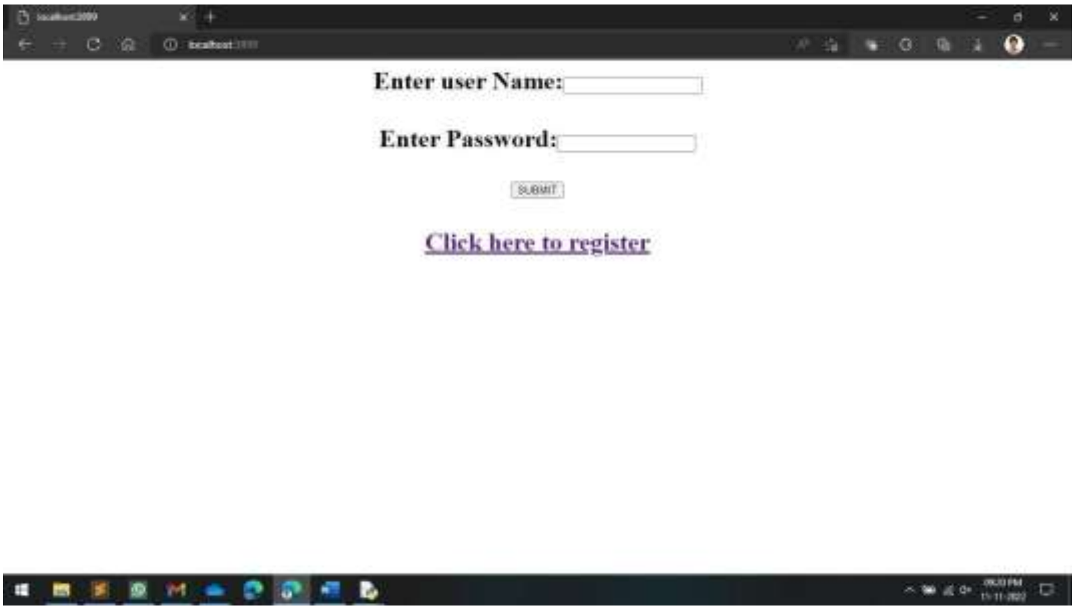
```
<html>
<body>
                              <center>
                   <img src="https://img.freepik.com/free-vector/flatdesigncolorful-characters-welcoming_23-2148271988.jpg?w=740&t=st=1668096317~exp=1668096917~hmac=da88964b5c0b6a1b878a26c38ba3a87abc6583421a79f1d4edac4abb2d71062e">
                              </center>
</body>
</html>
```

app.py:

```
from flask import
Flask,render_template,request,redirect,url_for,session import ibm_db
import re
app=Flask(__name__) app.secret_key
= 'abc'
conn=ibm_db.connect("DATABASE=bludb;HOSTNAME=ba99a9e6-d59e-4883-8fc0d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=31321;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=*************;PWD=************",'','') #Answer for Question(5)

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

```
def      home():                               return
render_template('login.html')

@app.route('/regis') def regis():     return
render_template('register.html')

@app.route('/login',methods=['GET','POST']) def
login():     global userid     msg=' '     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:
           msg='Logged in successfully!'            return
render_template('welcome.html',msg=msg)         else:
           return render_template('login.html')

@app.route('/register',methods=['GET','POST'])
def register():         if request.method=='POST':
username = request.form['username']            email
= request.form['email']            password
= request.form['password']
           sql = "SELECT * FROM User 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:                return
'{}'.format("Account already exist!")
else:
```

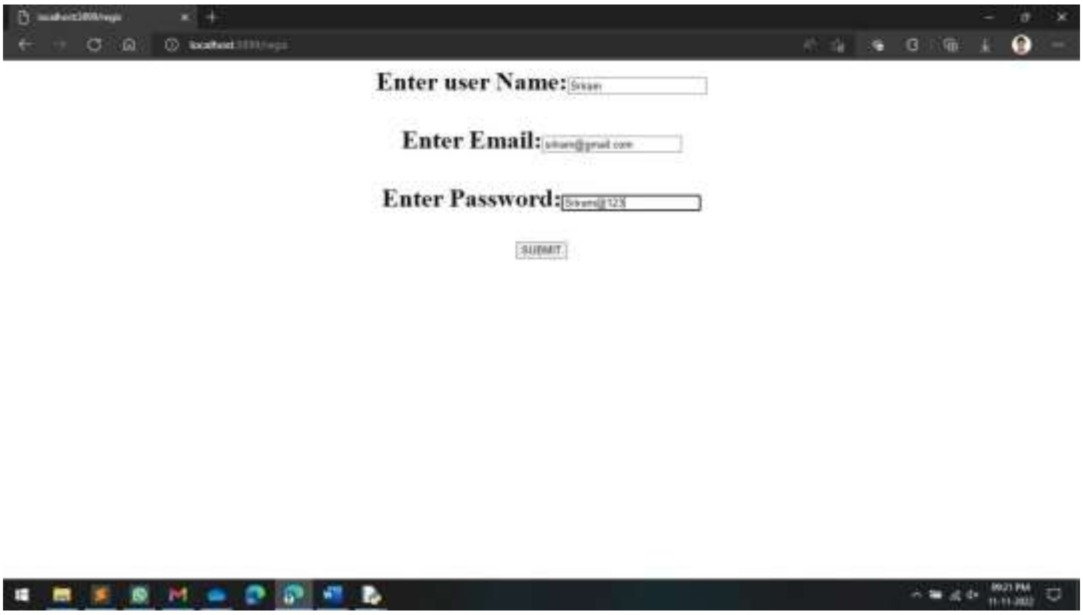
```
              insert_sql="INSERT INTO user VALUES(?, ?,
?)"              prep_stmt=ibm_db.prepare(conn,insert_sql)
ibm_db.bind_param(prep_stmt,1,username)
ibm_db.bind_param(prep_stmt,2,email)
ibm_db.bind_param(prep_stmt,3,password)
ibm_db.execute(prep_stmt)              msg="You have
successfully registered"              return render_template('login.html',msg=msg)

if __name__ == '__main__':
app.run('127.0.0.1',3899)
```



Enter user Name:

Enter Password:

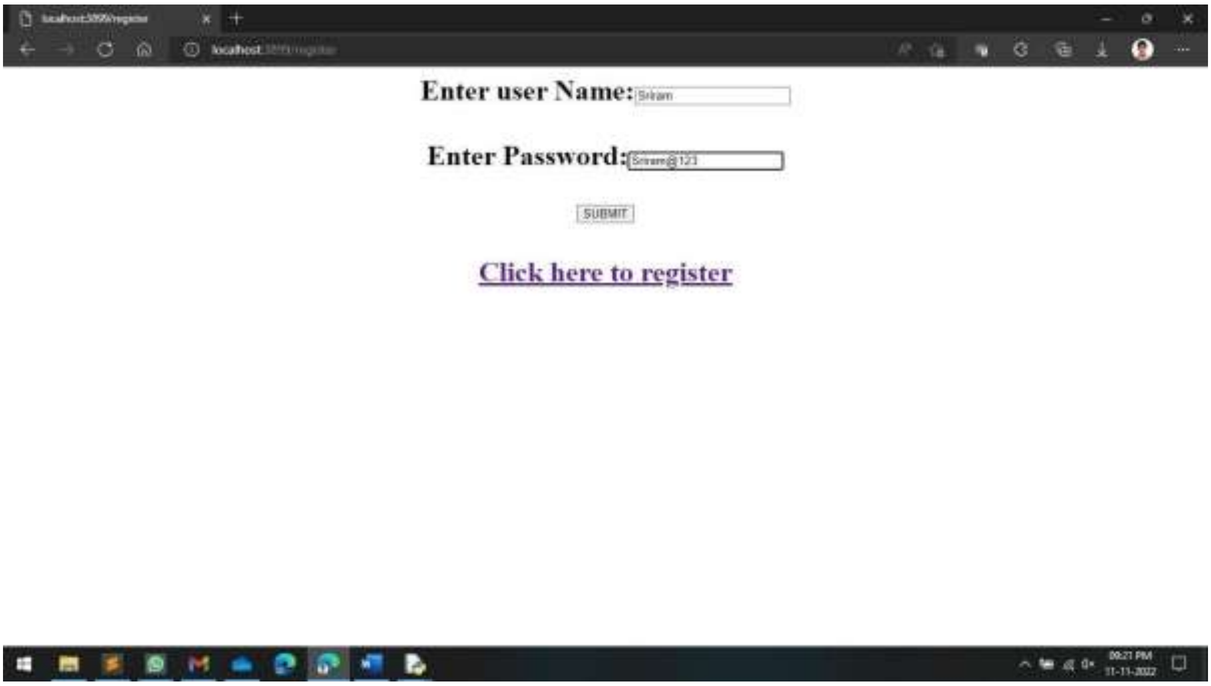
Click here to register



Enter user Name: Sriram

Enter Email: sriram@gmail.com

Enter Password: Sriram@123

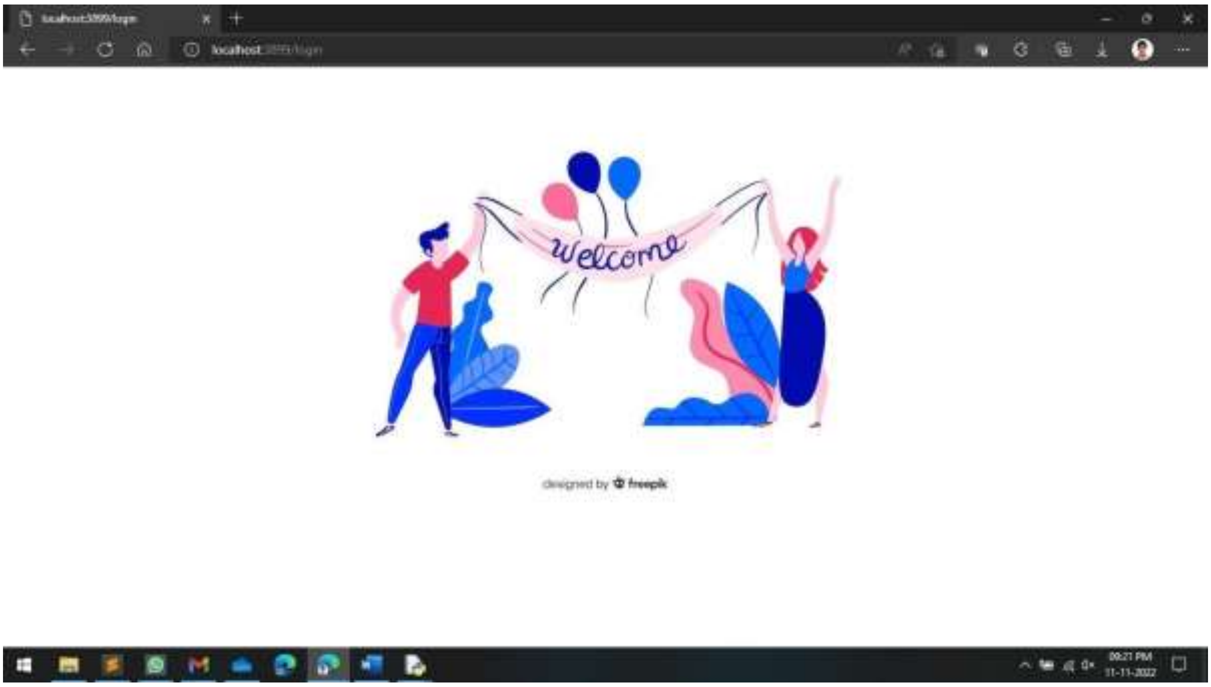


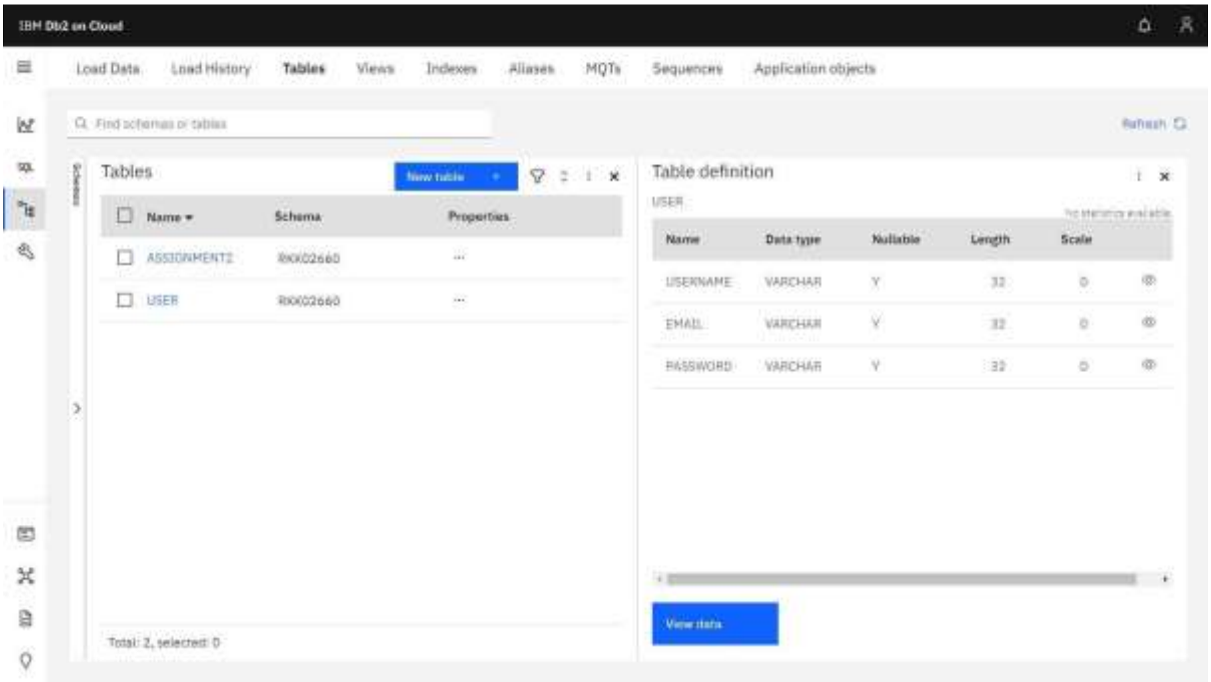
Enter user Name: Sriram

Enter Password: Sriram@123

SUBMIT

Click here to register



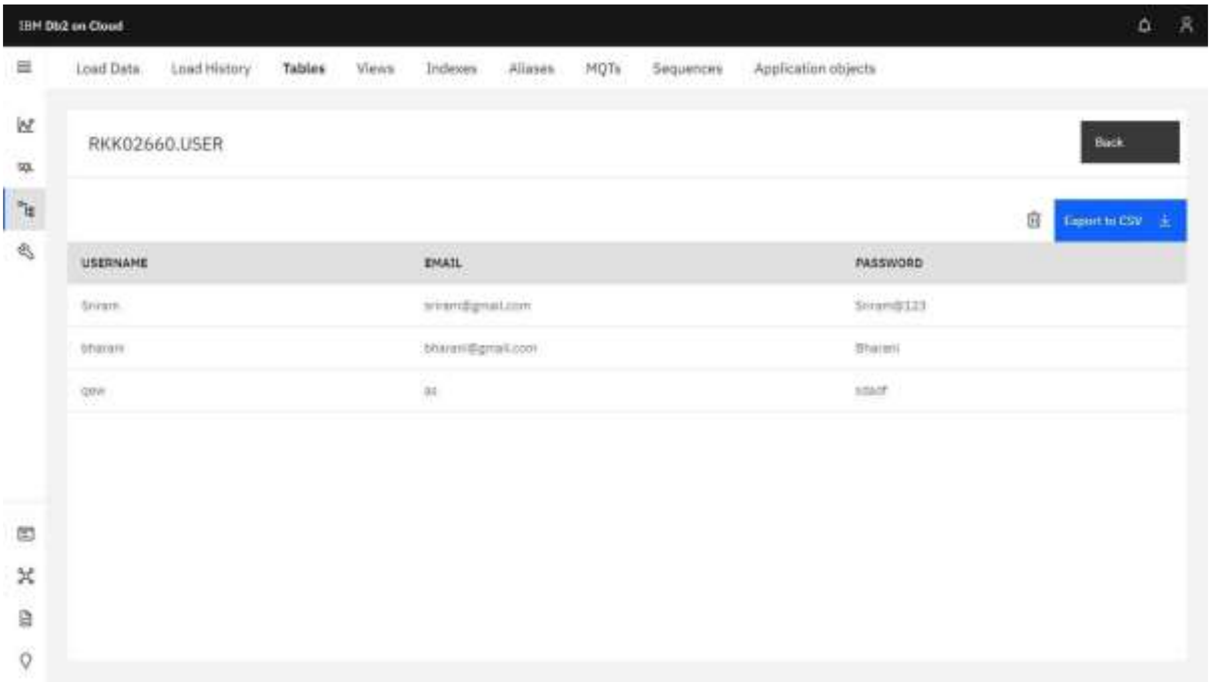


☐	Name ▾	Schema	Properties
☐	ASSIGNMENT2	RKK02660	···
☐	USER	RKK02660	···

Total: 2, selected: 0

USER

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



RKK02660.USER

USERNAME	EMAIL	PASSWORD
Sriram	sriram@gmail.com	Sriram@123
bharani	bharani@gmail.com	Bharani
qew	as	sdadf
