***Assignment-2***

|  |  |
| --- | --- |
| **Project Domain** | Cloud Application Development |
| **Project Title** | Customer Care Registry |
| **Team ID** | PNT2022TMID44404 |
| **Name** | BHARANIDHARAN B |
| **Roll No** | 731119205004 |
| **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
4. Connect python code to db2.
5. 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>

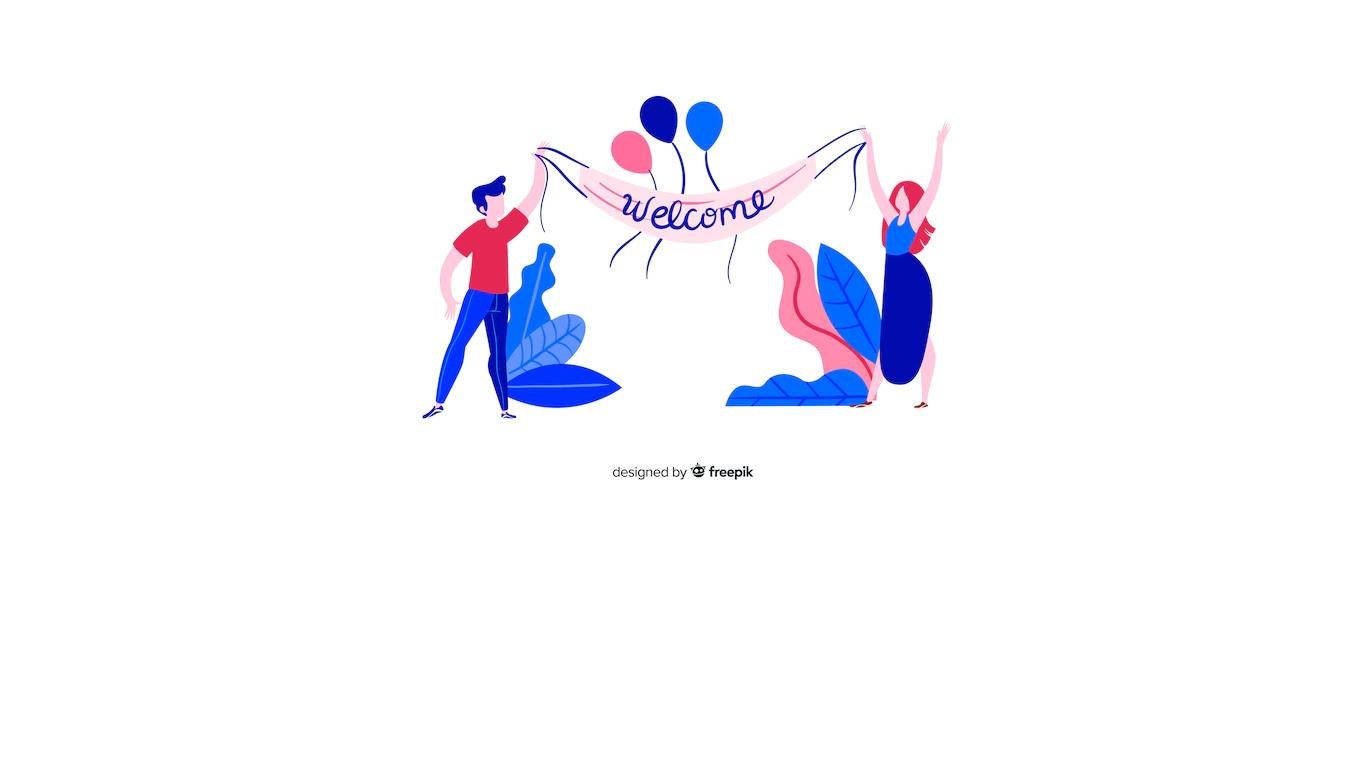
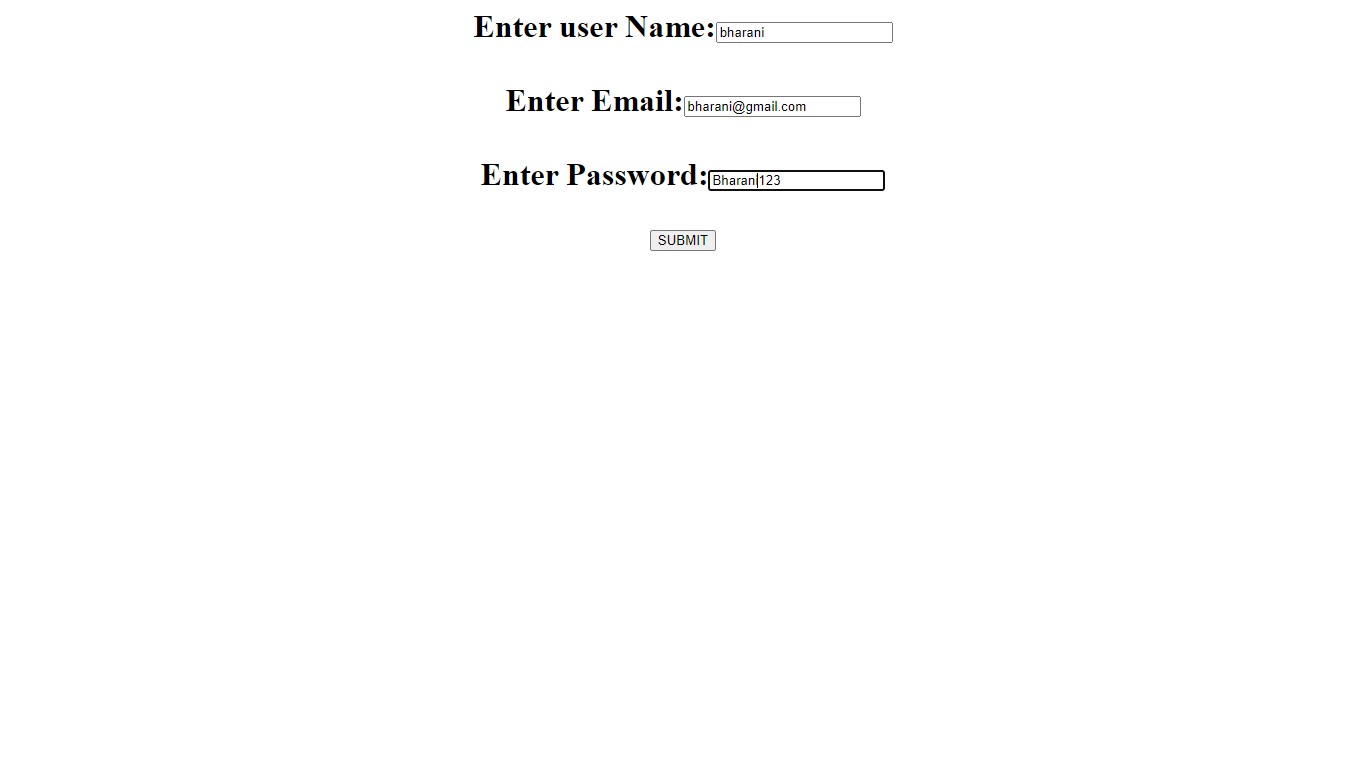
Enter Phone Number:<input type = "text" name "phoneNumber"/><br><br>

<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) |



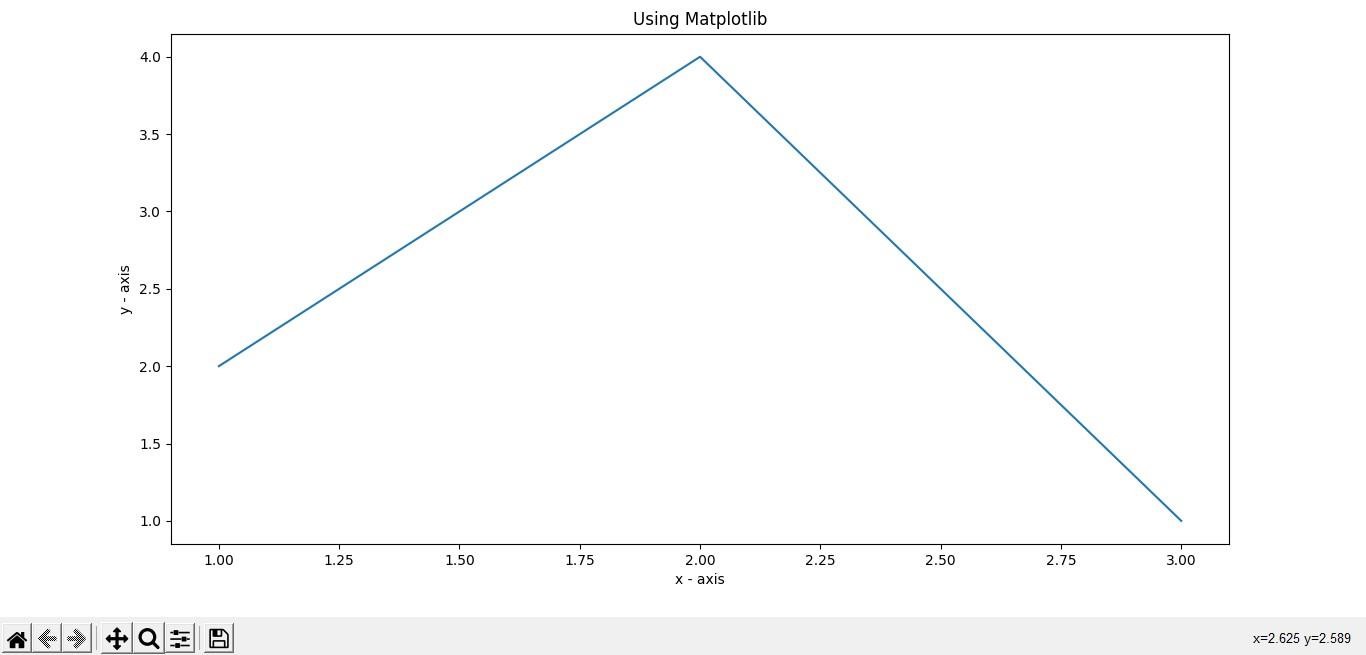
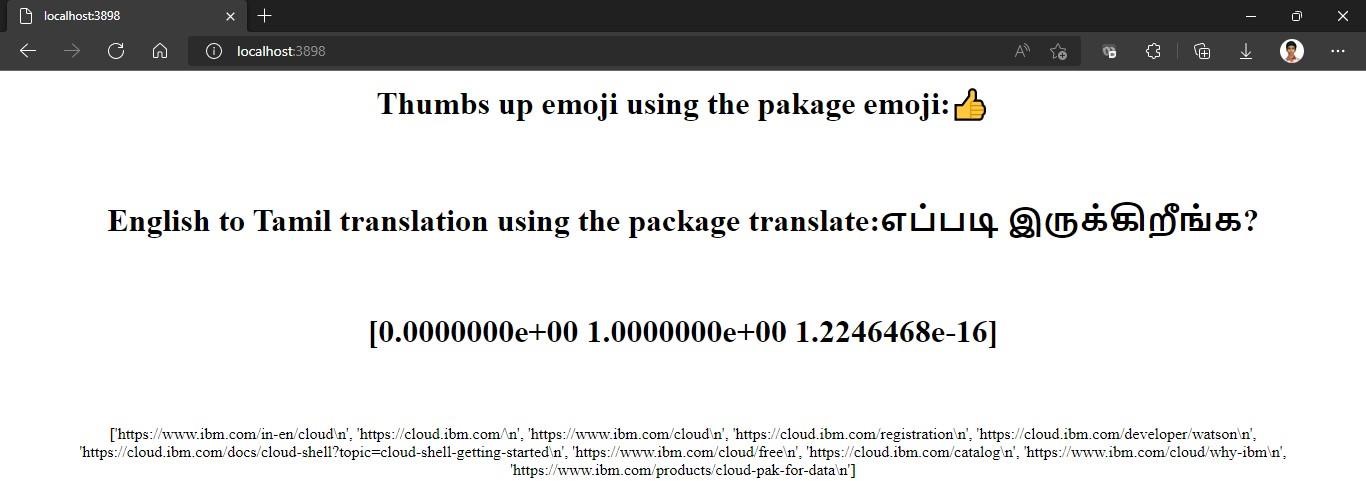
# ------------------------------------------------------------------------------------------------

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

***Packages used:***

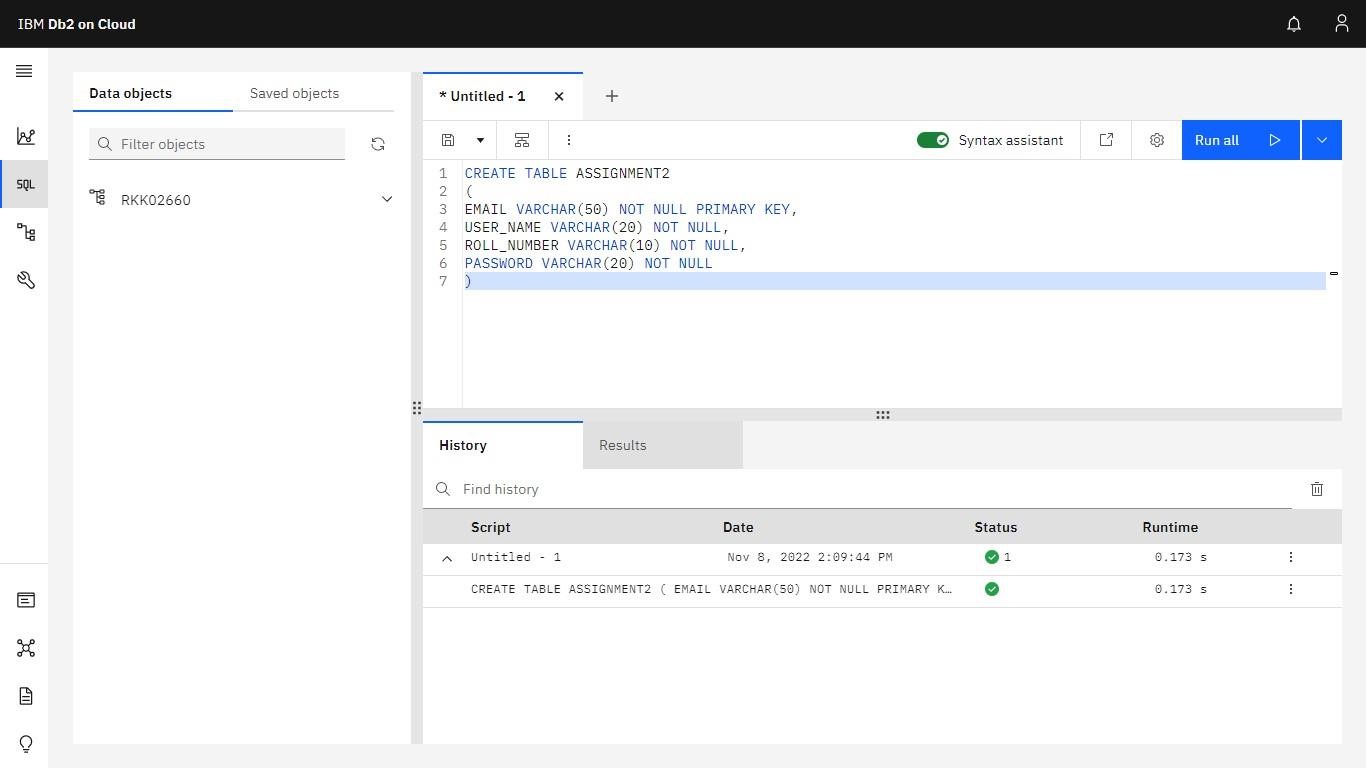
Flask, emoji, matplotlib, numpy, translate, googlesearch

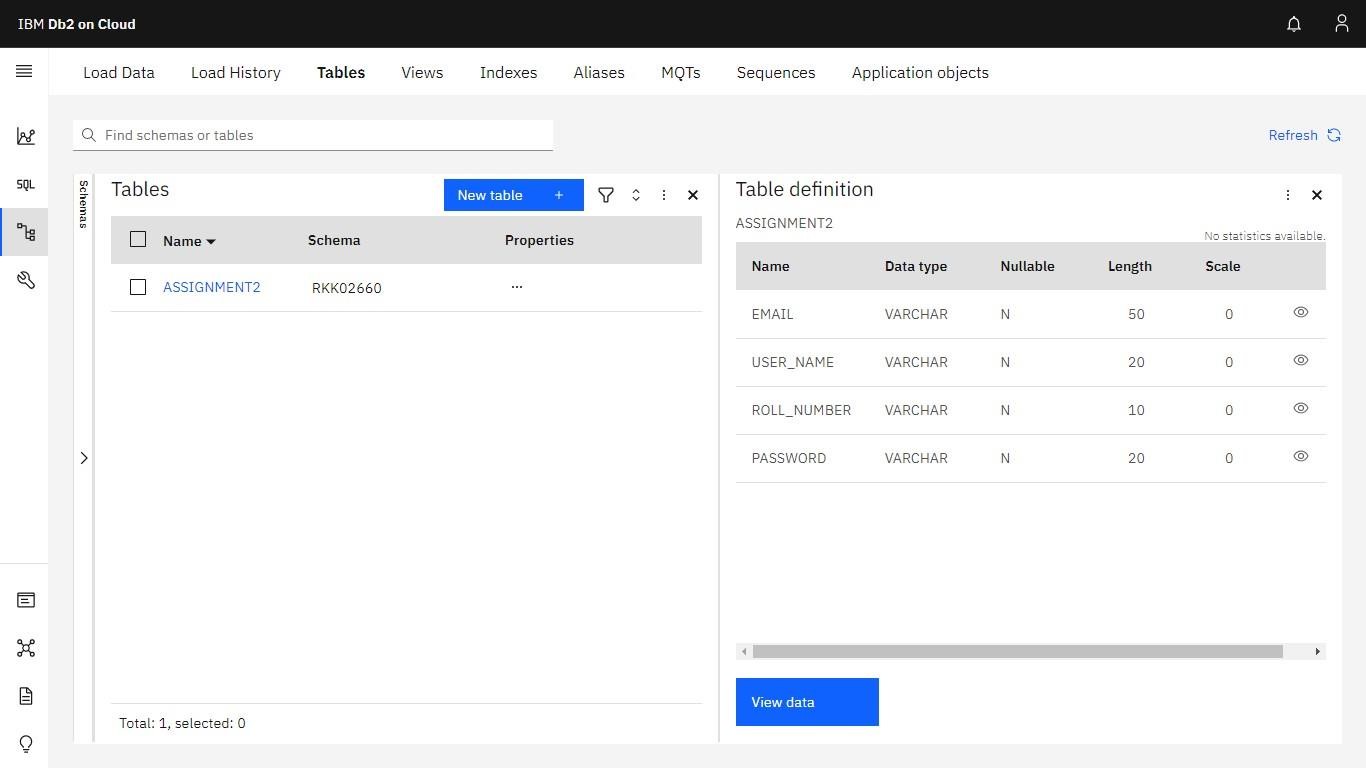
|  |
| --- |
| **Packages.py:**  from flask import Flask from emoji import emojize import matplotlib.pyplot as plt import numpy as np from translate import Translator  from googlesearch import search    app=Flask(\_\_name\_\_)    @app.route('/') def login(): emojione=("Thumbs up emoji using the pakage 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) |

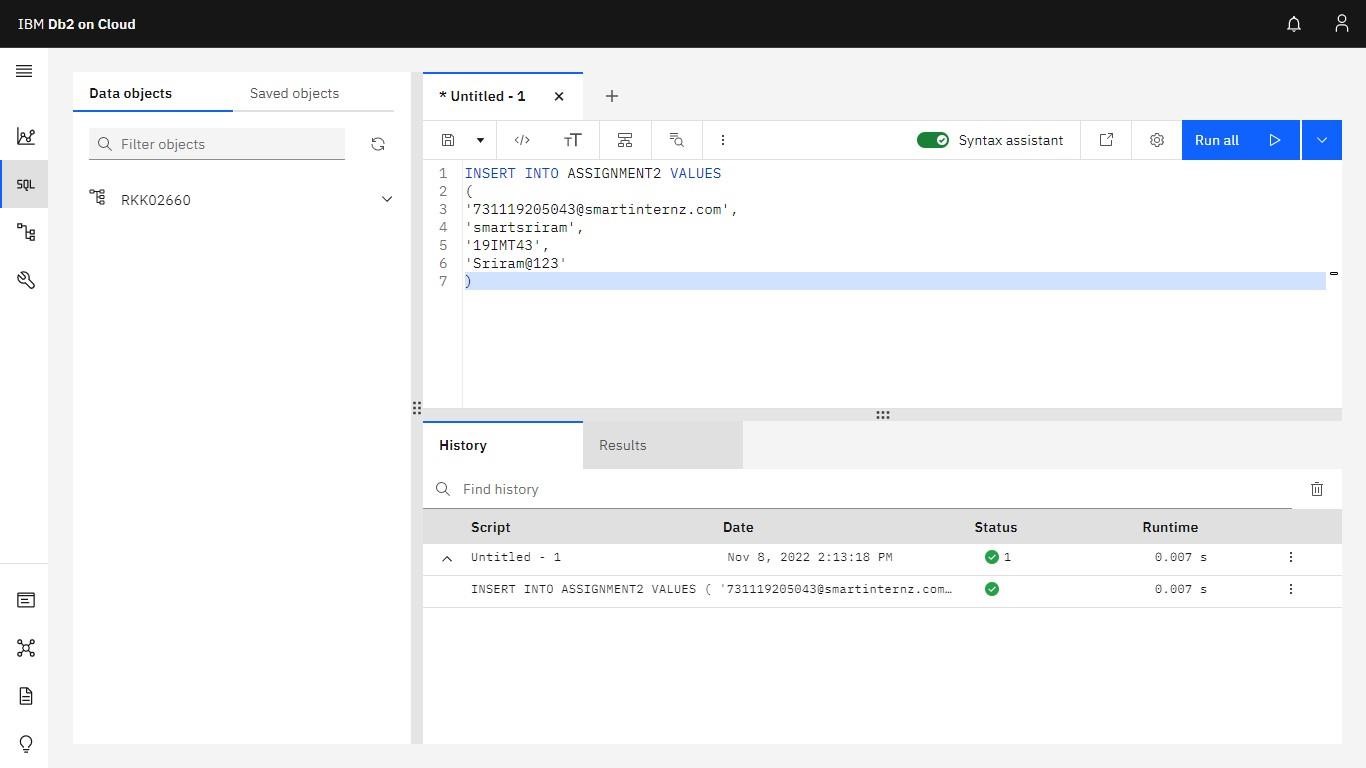


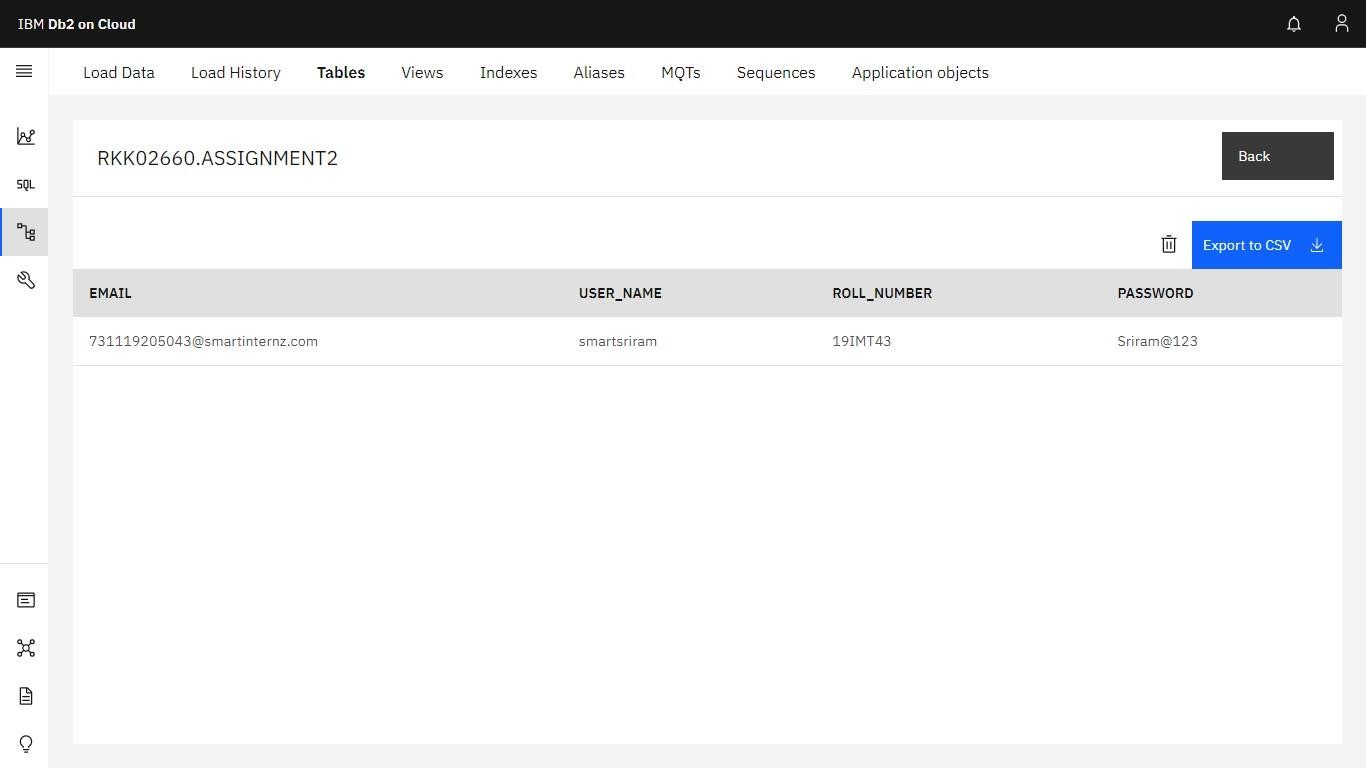
# ------------------------------------------------------------------------------------------------

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

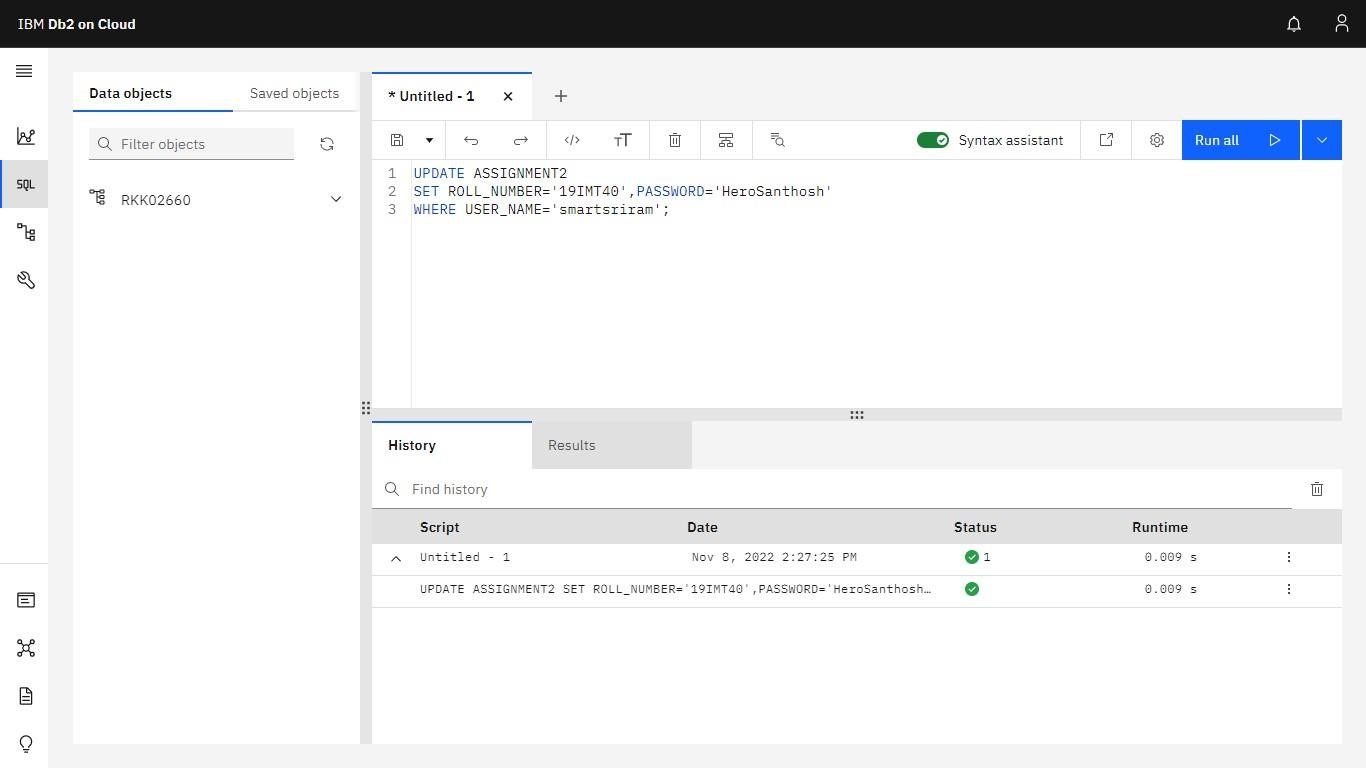


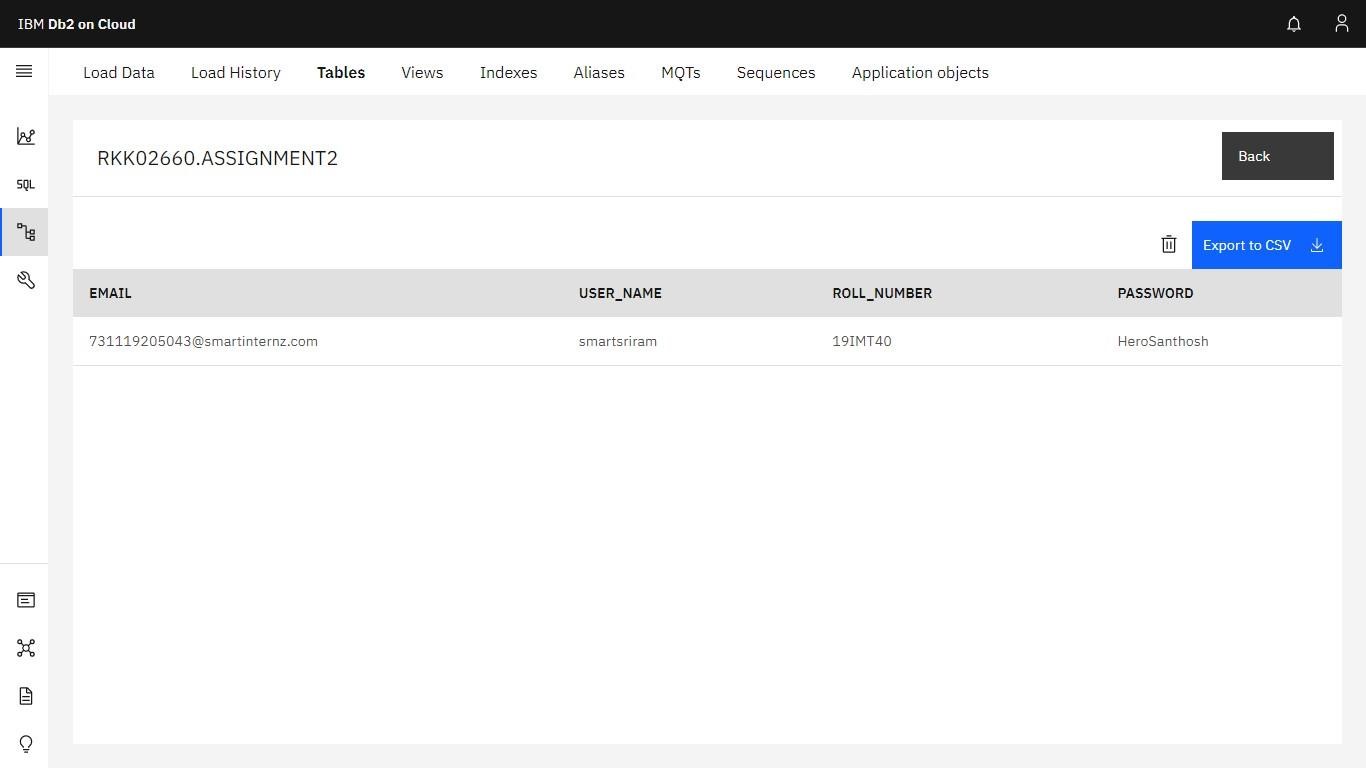


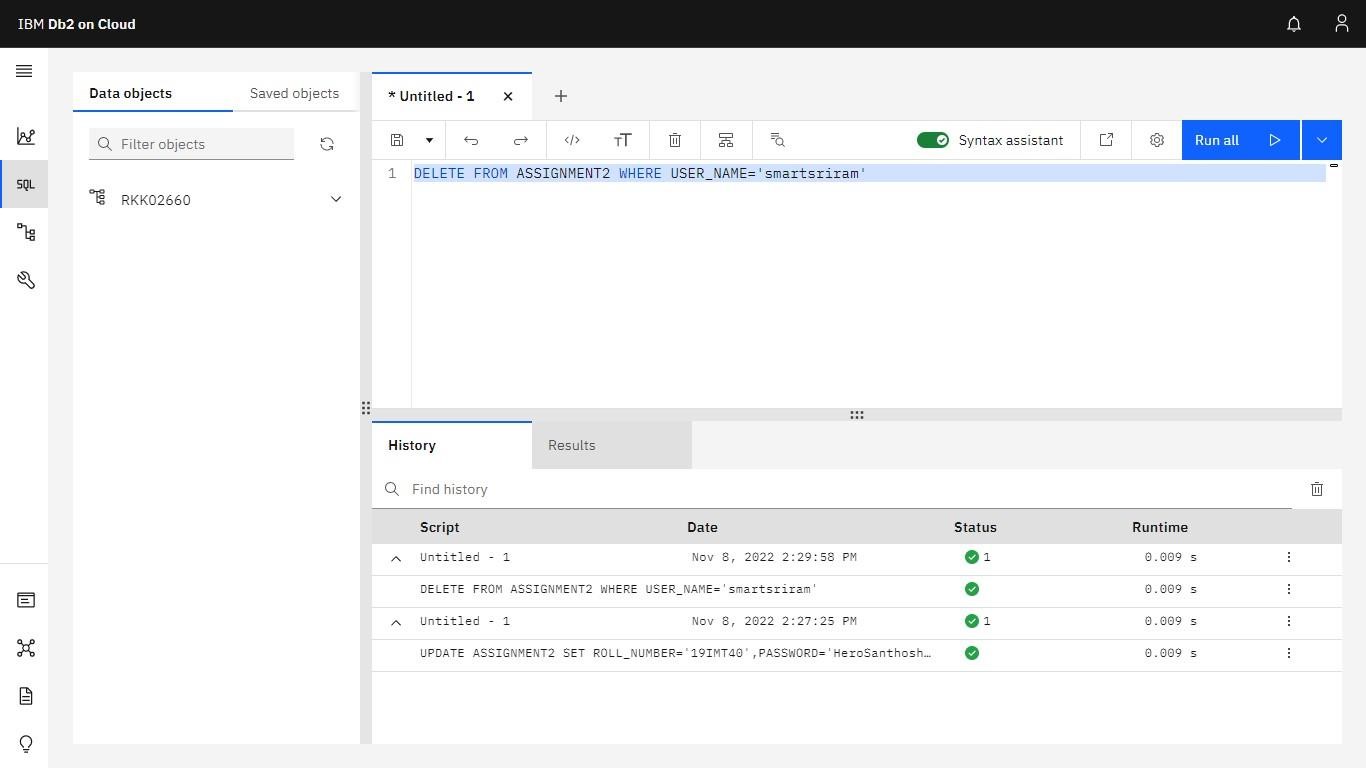


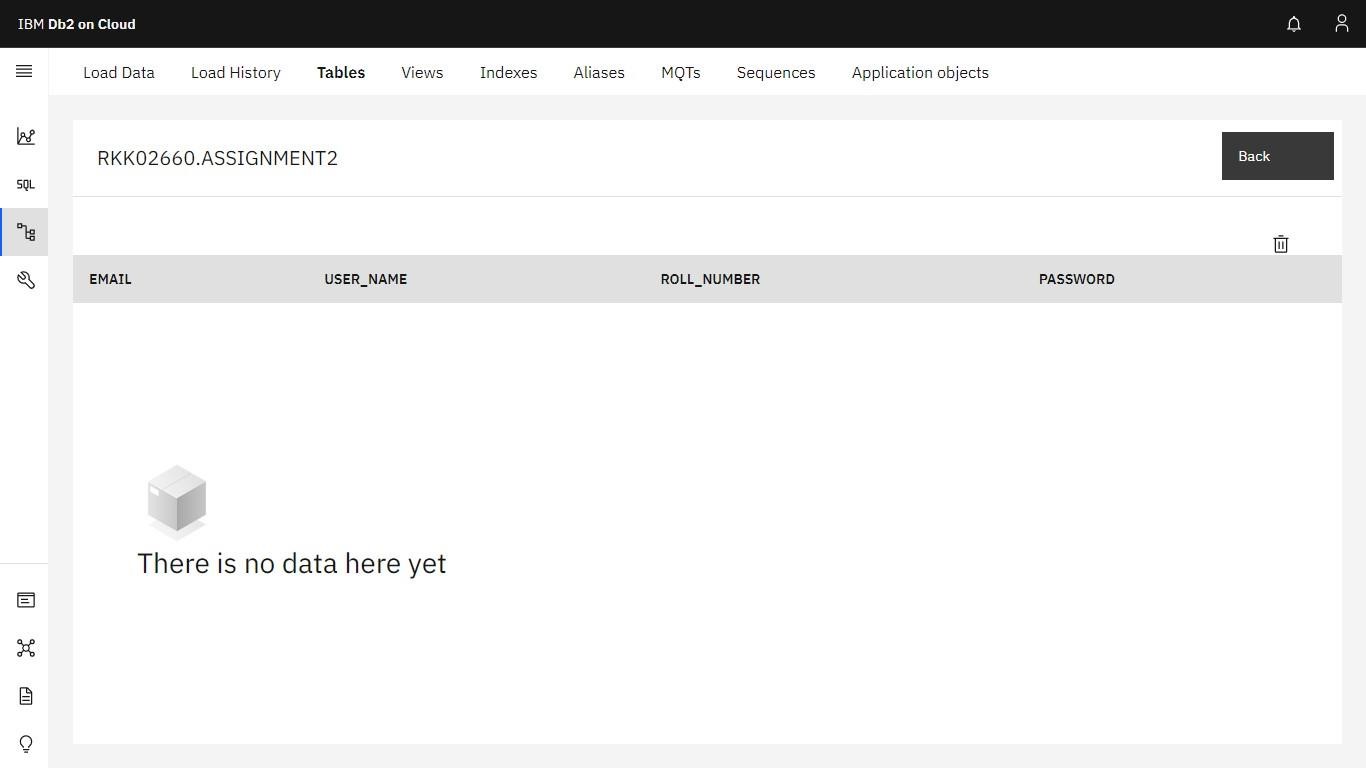


# ------------------------------------------------------------------------------------------------ 4.Perform UPDATE, DELETE Queries with user table









***------------------------------------------------------------------------------------------------***

1. Connect python code to db2.
2. 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=da8896  4b5c0b6a1b878a26c38ba3a87abc6583421a79f1d4edac4abb2d71062e">  </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=3 132

1;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(?, ?,

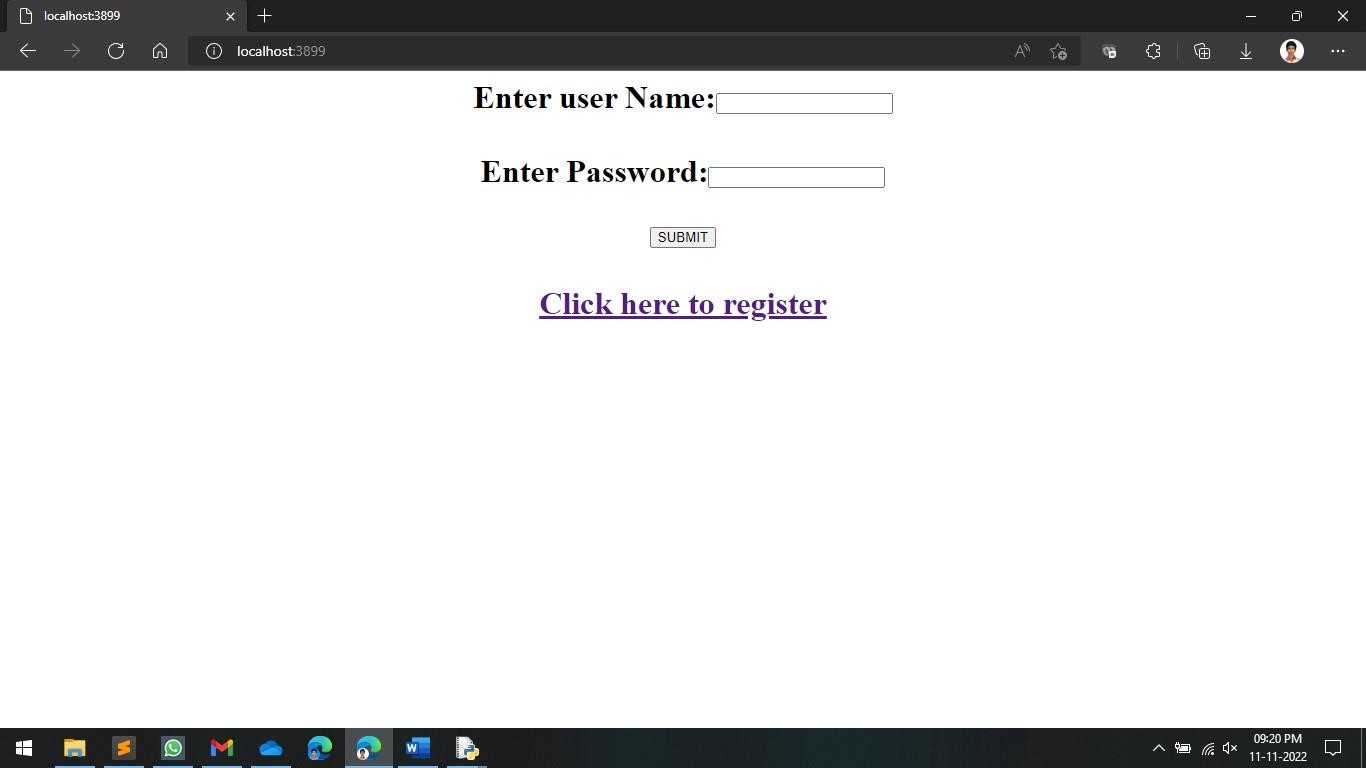
?)" prep\_stmt=ibm\_db.prepare(conn,insert\_sql)

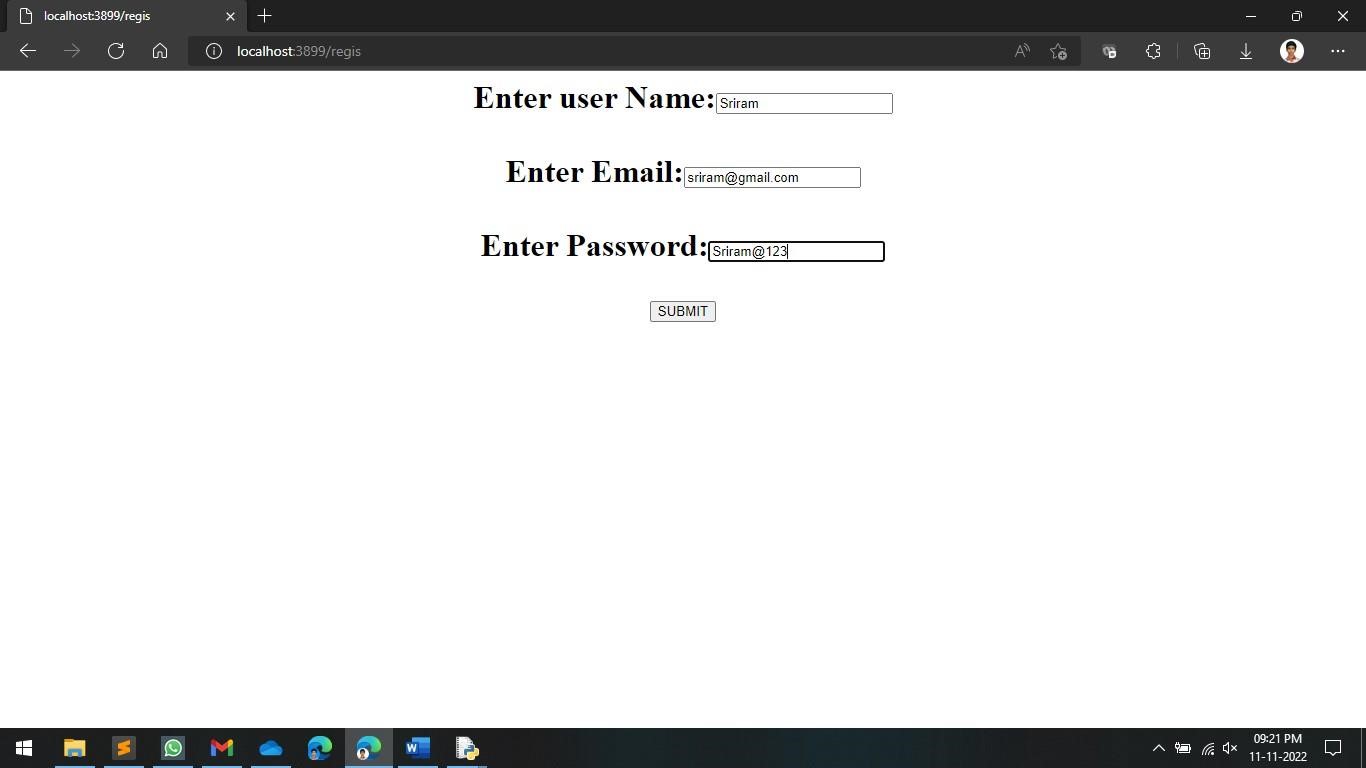
ibm\_db.bind\_param(prep\_stmt,1,username) ibm\_db.bind\_param(prep\_stmt,2,email) ibm\_db.bind\_param(prep\_stmt,3,password) ibm\_db.execute(prep\_stmt) msg="You have

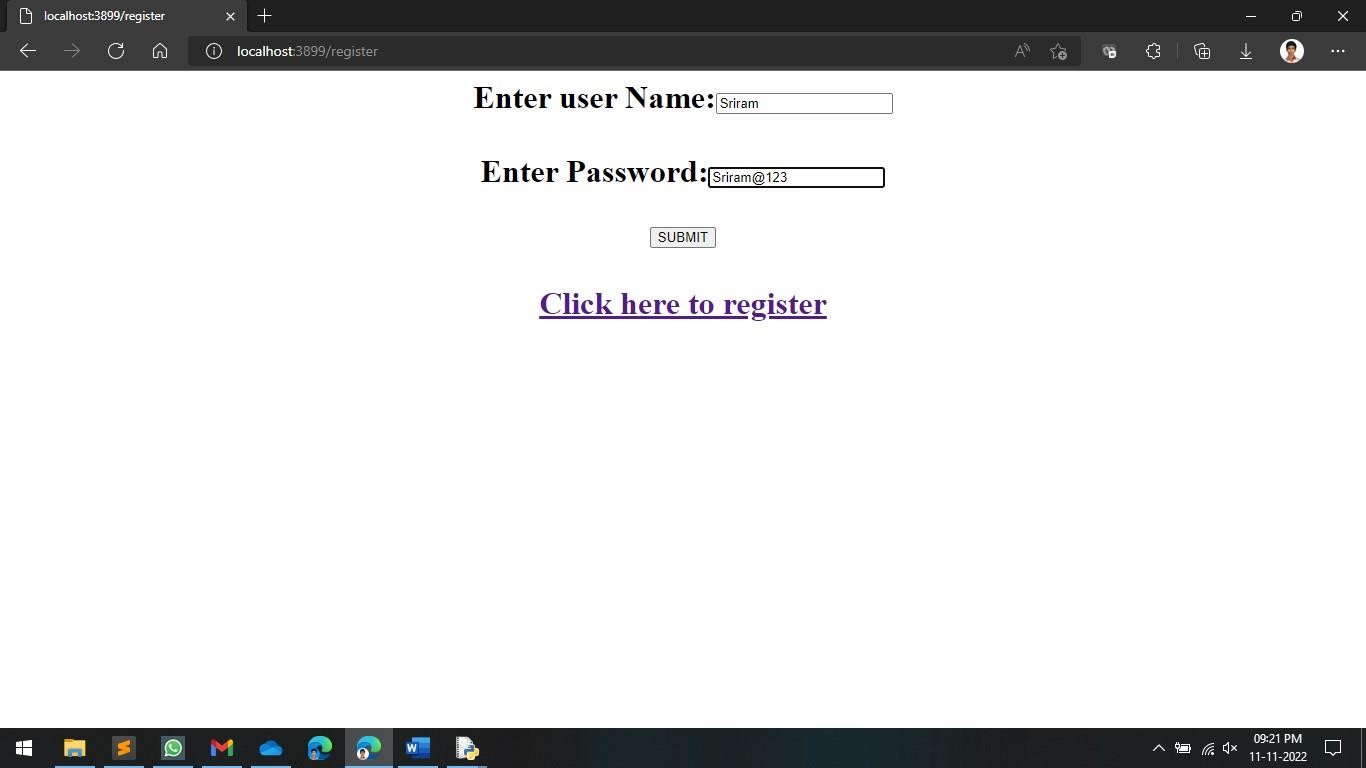
successfully registered" return render\_template('login.html',msg=msg)

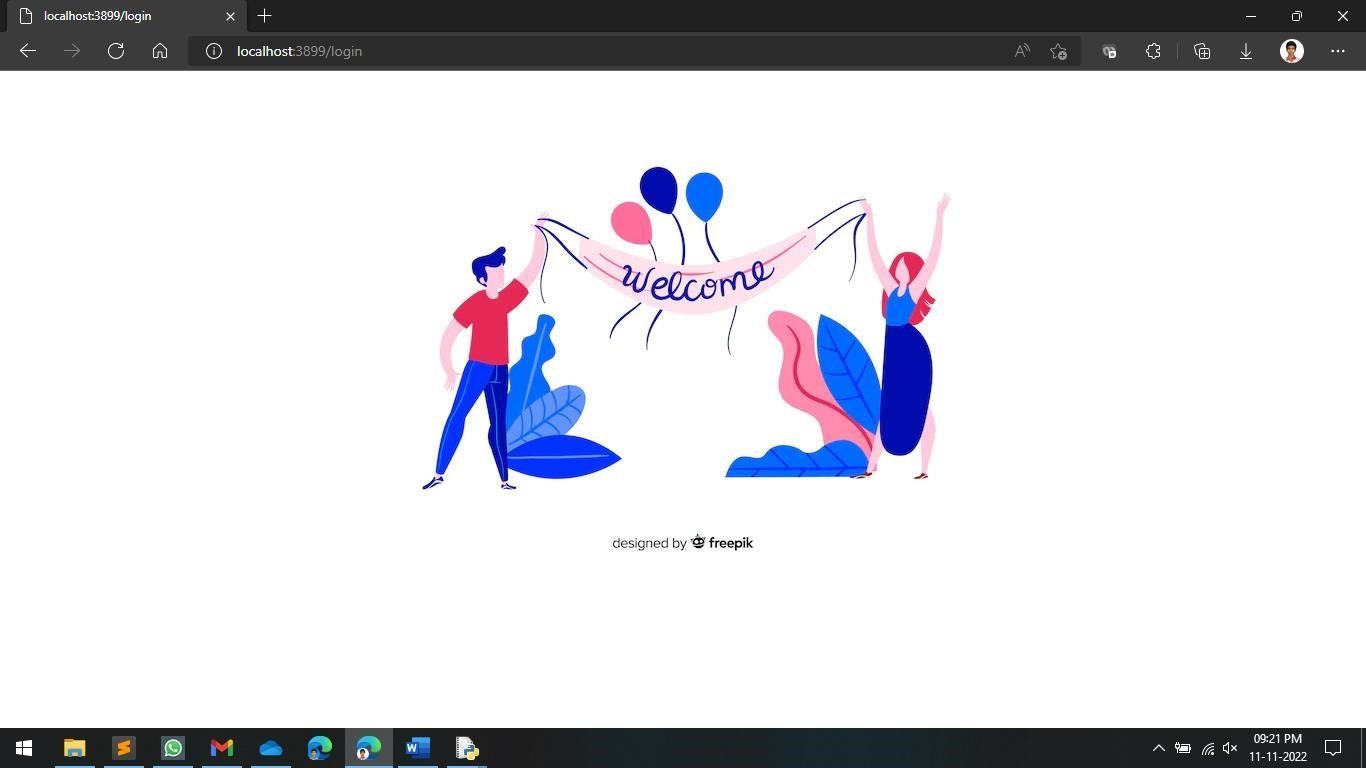
if \_\_name\_\_ == '\_\_main\_\_':

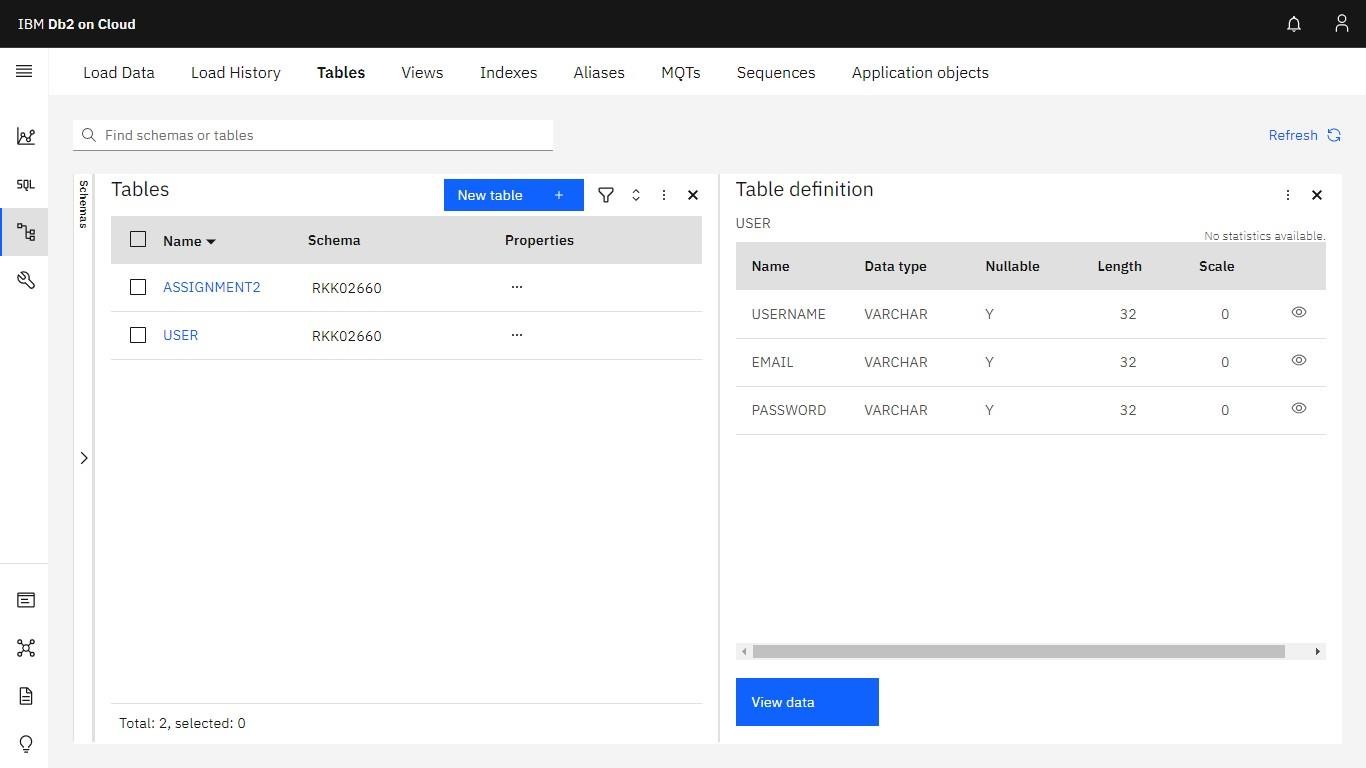
app.run('127.0.0.1',3899)

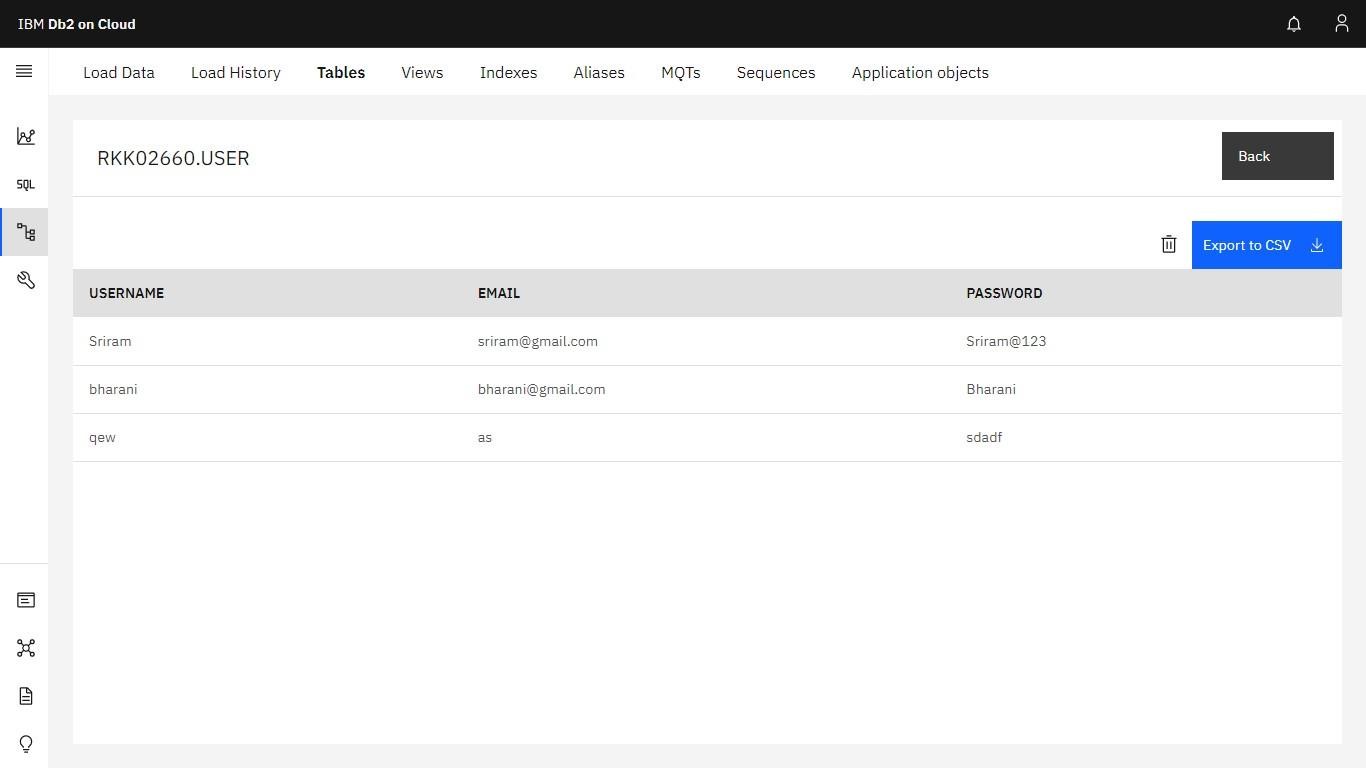












# ------------------------------------------------------------------------------------------------