

ASSIGNMENT - 2

Assignment Date	22 September 2022
Student Name	Jose Louie Shereen K
Student Roll Number	95071915019
Maximum Marks	2 Marks

QUESTION:

1. Create User table with user with email, username, roll number, password.
2. Perform UPDATE, DELETE Queries with user table
3. Connect python code to db2.
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.

1. CREATE USER TABLE WITH USER WITH EMAIL, USERNAME, ROLL NUMBER, PASSWORD.

The screenshot shows the IBM Db2 on Cloud web interface. The 'Tables' tab is selected, displaying a list of tables. The 'USER' table is selected, and its definition is shown in the 'Table definition' panel.

Name	Data type	Nullable	Length	Scale
ROLLNUMBER	INTEGER	Y		0
USERNAME	CHAR	Y	32	0
EMAIL	CHAR	Y	64	0
PASSWORD	VARCHAR	Y	32	0

Buttons: New table, View data

The screenshot shows the IBM Db2 on Cloud web interface. The 'YGT74733.USER' table is selected, and the 'Table definition' panel is visible. The table is empty, and a message indicates that there is no data yet.

ROLLNUMBER	USERNAME	EMAIL	PASSWORD
------------	----------	-------	----------

Message: There is no data here yet

IBM Db2 on Cloud

File Edit View History Bookmarks Tools Help

IBM Db2 on Cloud

Filter objects

SQL

YGT74733

Tables

USER

Views

MQTs

Aliases

Nicknames

*Untitled - 1

Syntax assistant

Run all

History

Find history

Script	Date	Status	Runtime
Untitled - 1	Oct 15, 2022 8:38:42 PM	2	0.047 s
INSERT INTO USER VALUES (1,'Shereen' , 'shereen@gmail.com' , 'Lucky7')			0.023 s
INSERT INTO USER VALUES (3,'Jose' , 'jose@gmail.com' , 'Jose32')			0.024 s

IBM Db2 on Cloud

File Edit View History Bookmarks Tools Help

IBM Db2 on Cloud

Load Data Load History Tables Views Indexes Aliases MQTs Sequences Application objects

YGT74733.USER

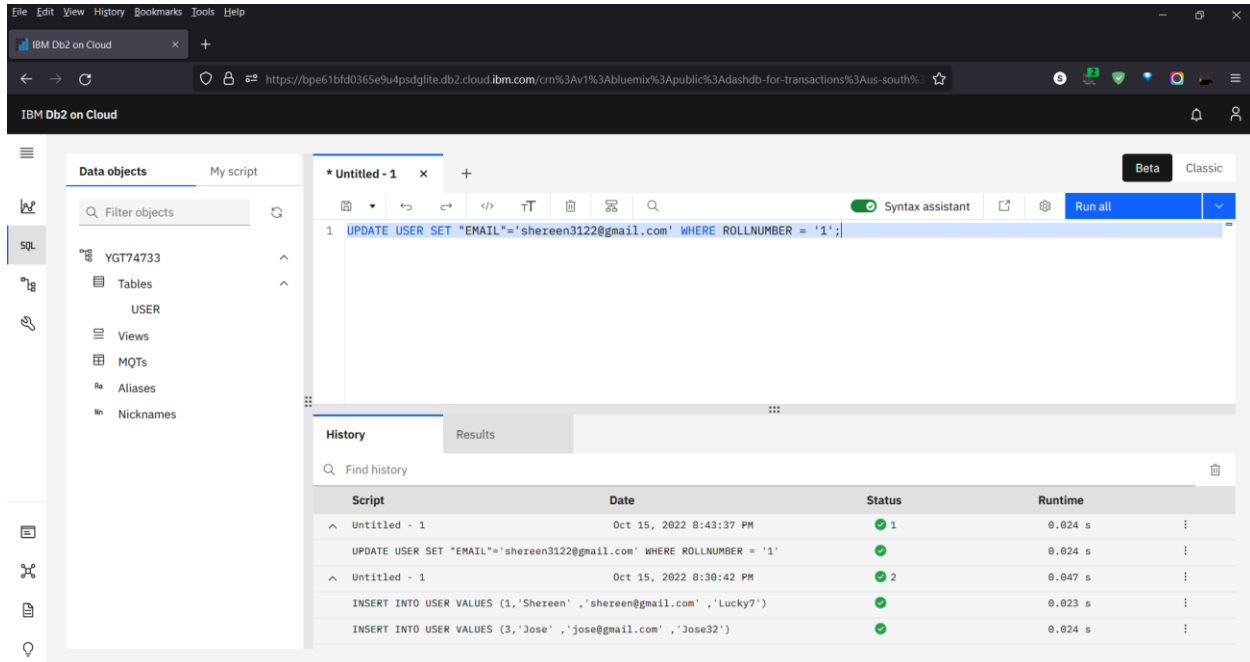
Back

Export to CSV

ROLLNUMBER	USERNAME	EMAIL	PASSWORD
1	Shereen	shereen@gmail.com	Lucky7
3	Jose	jose@gmail.com	Jose32

2. PERFORM UPDATE, DELETE QUERIES WITH USER TABLE.

UPDATE TABLE :

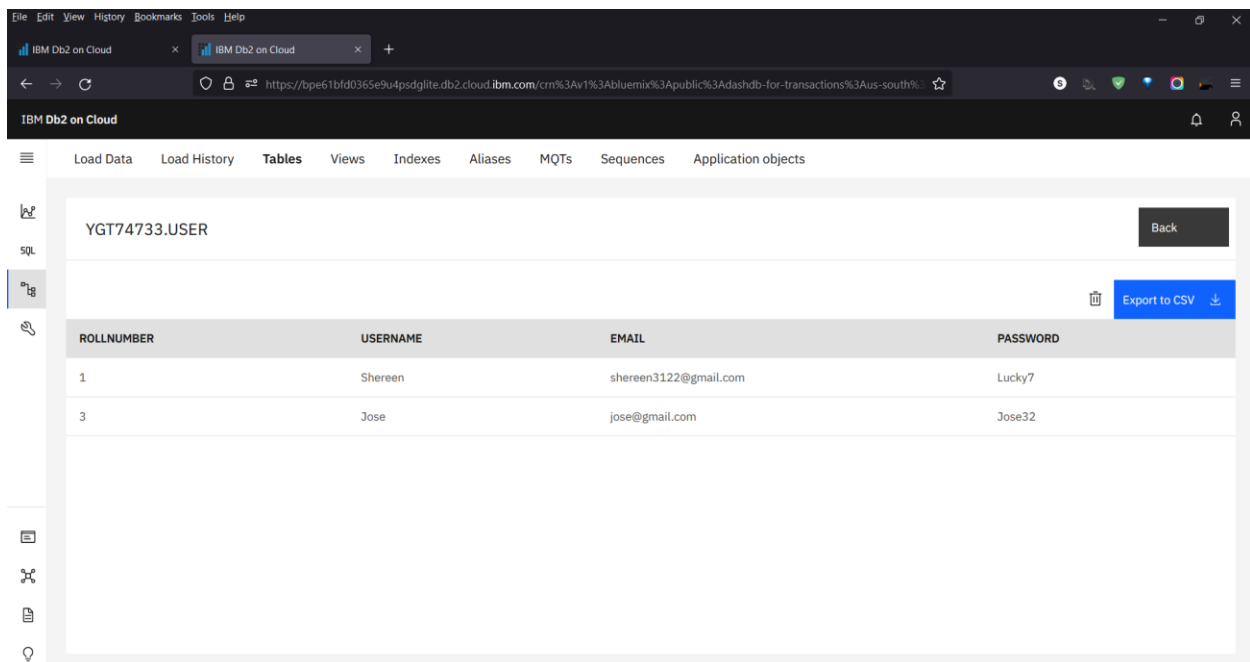


The screenshot shows the IBM Db2 on Cloud SQL editor interface. On the left, the 'Data objects' pane shows a tree structure with 'YGT74733' expanded, containing 'Tables', 'USER', 'Views', 'MQTs', 'Aliases', and 'Nicknames'. The main editor area shows a script titled 'Untitled - 1' with the following SQL query:

```
1 UPDATE USER SET "EMAIL"='shereen3122@gmail.com' WHERE ROLLNUMBER = '1';
```

Below the editor, the 'History' tab is active, displaying a table of executed scripts:

Script	Date	Status	Runtime
^ Untitled - 1	Oct 15, 2022 8:43:37 PM	✓ 1	0.024 s
UPDATE USER SET "EMAIL"='shereen3122@gmail.com' WHERE ROLLNUMBER = '1'		✓	0.024 s
^ Untitled - 1	Oct 15, 2022 8:30:42 PM	✓ 2	0.047 s
INSERT INTO USER VALUES (1,'Shereen' , 'shereen@gmail.com' , 'Lucky7')		✓	0.023 s
INSERT INTO USER VALUES (3,'Jose' , 'jose@gmail.com' , 'Jose32')		✓	0.024 s



The screenshot shows the IBM Db2 on Cloud SQL editor interface with the 'Tables' tab selected. The table 'YGT74733.USER' is displayed. The table structure is as follows:

ROLLNUMBER	USERNAME	EMAIL	PASSWORD
1	Shereen	shereen3122@gmail.com	Lucky7
3	Jose	jose@gmail.com	Jose32

Thus, for ROLL NUMBER 1 the EMAIL has been updated from "shereen@gmail.com" to "shereen3122@gmail.com".

DELETE TABLE :

The screenshot shows the IBM Db2 on Cloud SQL editor interface. On the left, the 'Data objects' pane shows a tree structure for 'YGT74733' containing 'Tables', 'USER', 'Views', 'MQTs', 'Aliases', and 'Nicknames'. The main editor area shows a script titled 'Untitled - 1' with the following SQL statement:

```
1 DELETE FROM USER WHERE ROLLNUMBER = '3';
```

Below the script, the 'History' tab is active, displaying a table of executed queries:

Script	Date	Status	Runtime
Untitled - 1	Oct 15, 2022 8:46:44 PM	✓ 1	0.007 s
DELETE FROM USER WHERE ROLLNUMBER = '3'		✓	0.007 s
Untitled - 1	Oct 15, 2022 8:43:37 PM	✓ 1	0.024 s
UPDATE USER SET 'EMAIL'='shereen3122@gmail.com' WHERE ROLLNUMBER = '1'		✓	0.024 s
Untitled - 1	Oct 15, 2022 8:30:42 PM	✓ 2	0.047 s
INSERT INTO USER VALUES (1,'Shereen' , 'shereen@gmail.com' , 'Lucky7')		✓	0.023 s
INSERT INTO USER VALUES (3,'Jose' , 'jose@gmail.com' , 'Jose32')		✓	0.024 s

The screenshot shows the IBM Db2 on Cloud SQL editor interface. The 'Data objects' pane on the left shows the 'YGT74733.USER' table selected. The main editor area displays the table's data:

ROLLNUMBER	USERNAME	EMAIL	PASSWORD
1	Shereen	shereen3122@gmail.com	Lucky7

Buttons for 'Back' and 'Export to CSV' are visible in the top right corner of the table view.

Thus ROLLNUMBER 2 has been deleted.

3. CONNECT PYTHON CODE TO DB2.

* NOTE:- Question 4 Contains Question 3 Answer

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

```
1 from flask import Flask, render_template, request, redirect, url_for
2 import ibm_db
3
4 dsn_hostname = "b1bc1829-6f45-4cd4-bef4-10cf081900bf.clogj3sdtgtu0lqdc00.databases.appdomain.cloud"
5 dsn_uid = "bpu31478" # e.g. "abc12345"
6 dsn_pwd = "zKP9mhh8R2ufS8gT" # e.g. "7dRZ3hdt5XN6$00"
7 dsn_driver = "[IBMDBZCL1]"
8 dsn_database = "bludb"
9 dsn_port = "32304" # e.g. "32773"
10 dsn_protocol = "TCP/IP" # e.g. "TCP/IP"
11 dsn_security = "SSL" # i.e. "SSL"
12 dsn_cert = "DigiCertGlobalRootCA.crt"
13 dsn = ("DRIVER={0}"; "DATABASE={1}"; "HOSTNAME={2}"; "PORT={3}"; "PROTOCOL={4}"; "UID={5}"; "PWD={6}"; "SECURITY={7}"; "SSLServerCertificate={8}").format(dsn_driver, dsn_database, dsn_hostname, dsn_port, dsn_protocol, dsn_uid, dsn_pwd, dsn_security, dsn_cert)
14 conn = ibm_db.connect(dsn, "", "")
15 app = Flask(__name__)
16 @app.route("/", methods=['GET', 'POST'])
17 def regpage():
18     if request.method == 'POST':
19         USERNAME = request.form['username']
20         PHONE = request.form['phone']
21         PASSWORD = request.form['password']
22         CPASSWORD = request.form['cpassword']
23         insertQuery = "INSERT INTO USER VALUES ('"+USERNAME+"','"+PHONE+"','"+PASSWORD+"','"+CPASSWORD+"')"
24         insertStmt = ibm_db.prepare(conn, insertQuery)
25         ibm_db.execute(insertStmt)
26         return redirect(url_for('loginpage'))
27     return render_template("register.html")
28 @app.route("/login", methods=['GET', 'POST'])
29 def loginpage():
30     if request.method == 'POST':
31         phone = request.form['phone']
32         password = request.form['password']
33         query = "select COUNT(*) from user where phone='"+phone+"' and password='"+password'"
34         stmt = ibm_db.exec_immediate(conn, query)
35         row = ibm_db.fetch_tuple(stmt)
36         if(row[0] == 1):
37             return redirect(url_for('home_page'))
38         return render_template("login.html")
39
40 def home_page():
41     return render_template("dashboard.html")
42 if __name__ == '__main__':
43     app.run()
```

Register.html

```
1 <head>
2 <meta name="viewport" content="width=device-width, initial-scale=1.0">
3 <link rel="stylesheet" href="/static/style.css">
4 <title>Registration page </title>
5 </head>
6 <body>
7 <div class="container">
8 <div class="title">Registration</div>
9 <div class="content">
10 <form method="post">
11 <div class="user-details">
12 <div class="input-box">
13 <span class="details">Username</span>
14 <input type="text" placeholder="Enter your username" name="username" value="{{request.form['username']}}" required>
15 </div>
16 <div class="input-box">
17 <span class="details">Phone Number</span>
18 <input type="text" placeholder="Enter your number" name="phone" value="{{request.form['phone']}}" required>
19 </div>
20 <div class="input-box">
21 <span class="details">Password</span>
22 <input type="password" placeholder="Enter your password" name="password" value="{{request.form['password']}}" required>
23 </div>
24 <div class="input-box">
25 <span class="details">Confirm Password</span>
26 <input type="password" placeholder="Confirm your password" name="cpassword" value="{{request.form['cpassword']}}" required>
27 </div>
28 </div>
29 <div class="button"><input type="submit" value="Register"></div>
30 </form>
31 </div>
32 </div>
33 </body>
34 </html>
```

Login.html

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <meta http-equiv="X-UA-Compatible" content="IE=edge">
6 <meta name="viewport" content="width=device-width, initial-scale=1.0">
7 <link rel="stylesheet" href="/static/style.css">
8 <link rel="stylesheet" href="{{url_for('static',filename='style.css')}}" type="text/css">
9 <title>Login Page</title>
10 </head>
11 <body>
12 <div class="container">
13 <div class="title">login</div>
14 <div class="content">
15 <form method="post" action="login">
16 <div class="user-details">
17 <div class="input-box">
18 <span class="details">Phone Number</span>
19 <input type="text" placeholder="Enter your number" name="phone" value="{{request.form['phone']}}" required>
20 </div>
21 <br>
22 <div class="input-box">
23 <span class="details">Password</span>
24 <input type="password" placeholder="Enter your password" name="password" value="{{request.form['password']}}" required>
25 </div>
26 </div>
27 <div class="button">
28 <input type="submit" value="Login">
29 </div>
30 </form>
31 </div>
32 </div></body>
33 </html>
```

Style.css

```
1 *{
2   margin: 0;
3   padding: 0;
4   box-sizing: border-box;
5   font-family: 'Poppins', sans-serif;
6 }
7 body{
8   height: 100vh;
9   display: flex;
10  justify-content: center;
11  align-items: center;
12  padding: 10px;
13  background: linear-gradient(135deg, #71b7e6, #9b59b6);
14 }
15 .container{
16   max-width: 700px;
17   width: 100%;
18   background-color: #fff;
19   padding: 25px 30px;
20   border-radius: 5px;
21   box-shadow: 0 5px 10px rgba(0,0,0,0.15);
22 }
23 .container .title{
24   font-size: 25px;
25   font-weight: 500;
26   position: relative;
27 }
28 .container .title::before{
29   content: "";
30   position: absolute;
31   left: 0;
32   bottom: 0;
33   height: 3px;
34   width: 30px;
35   border-radius: 5px;
36   background: linear-gradient(135deg, #71b7e6, #9b59b6);
37 }
```

OUTPUT :-

Registration Page:-

Registration page

127.0.0.1:5000

Registration

Username

Shayu

Phone Number

9894222470

Password

.....

Confirm Password

.....

Register

Login Page:-

Login Page

127.0.0.1:5000/login

Login

Phone Number

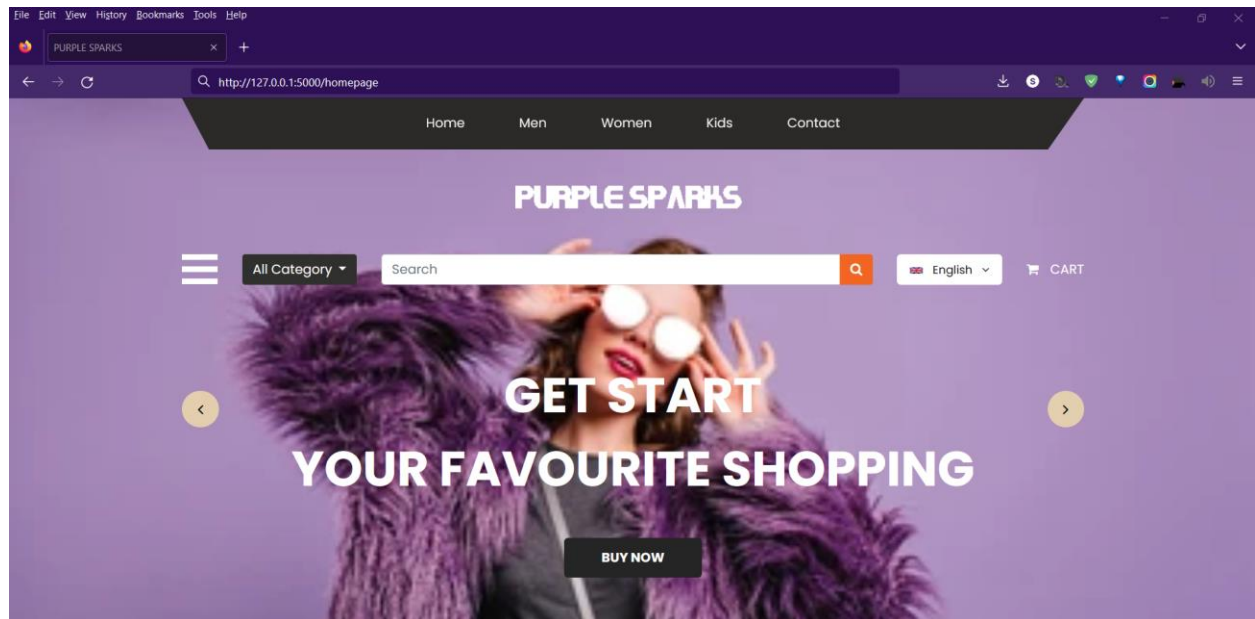
9894222470

Password

.....

Login

Welcome page:-



Database Table :-

