

ASSIGNMENT - 2

Assignment Date	22 September 2022
Student Name	Vimala P
Student Roll Number	95071915053
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 highlighted. The 'Table definition' panel on the right shows the table's structure:

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

The interface also shows a search bar, navigation tabs (Load Data, Load History, Tables, Views, Indexes, Aliases, MQTs, Sequences, Application objects), and a sidebar with icons for various database functions.

The screenshot shows the IBM Db2 on Cloud web interface, displaying the 'PDJ43917.USER' table. The table structure is shown at the top:

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

Below the table structure, a message states: "There is no data here yet". The interface includes a 'Back' button and a trash icon. The bottom of the screen shows the Windows taskbar with the system clock at 12:32 PM on 05-11-2022.

Service Details - IBM Cloud x IBM Db2 on Cloud x +

bpe61bfd0365e9u4psdglite.db2.cloud.ibm.com/crn%3Av1%3Abluemix%3Apublic%3Adashdb-for-transactions%3Aus-south%3Aa%2F5366ae6b3c5437892526264e84e3374%3A459e1966-85f6-412...

IBM Db2 on Cloud

Data objects

Saved objects

Filter objects

PDJ43917

*Untitled - 1

1 INSERT INTO USER VALUES(1,'Vimala','vimala18@gmail.com','vimala11@');

2 INSERT INTO USER VALUES(2,'Srimathi','srimathii12@gmail.com','sri12@');

History

Find history

Script	Date	Status	Runtime
Untitled - 1	Nov 5, 2022 12:35:17 PM	2	0.012 s
INSERT INTO USER VALUES(1,'Vimala','vimala18@gmail.com','vimala11@')			0.006 s
INSERT INTO USER VALUES(2,'Srimathi','srimathii12@gmail.com','sri12@')			0.006 s

Type here to search 31°C Cloudy 12:35 PM 05-11-2022

Service Details - IBM Cloud x IBM Db2 on Cloud x +

bpe61bfd0365e9u4psdglite.db2.cloud.ibm.com/crn%3Av1%3Abluemix%3Apublic%3Adashdb-for-transactions%3Aus-south%3Aa%2F5366ae6b3c5437892526264e84e3374%3A459e1966-85f6-412...

IBM Db2 on Cloud

Load Data

Load History

Tables

Views

Indexes

Aliases

MQTs

Sequences

Application objects

PDJ43917.USER

Back

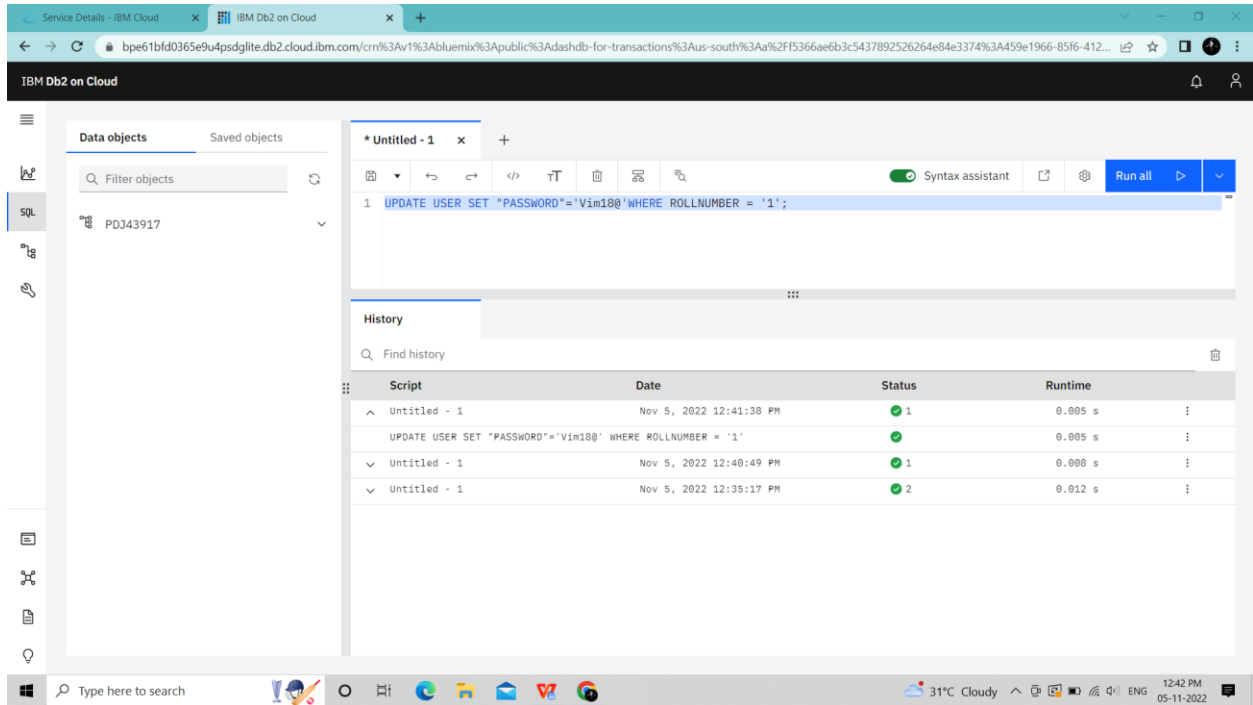
Export to CSV

ROLLNUMBER	USERNAME	EMAIL	PASSWORD
1	Vimala	vimala18@gmail.com	vimala11@
2	Srimathi	srimathii12@gmail.com	sri12@

Type here to search 31°C Cloudy 12:38 PM 05-11-2022

2. PERFORM UPDATE, DELETE QUERIES WITH USER TABLE.

UPDATE TABLE :

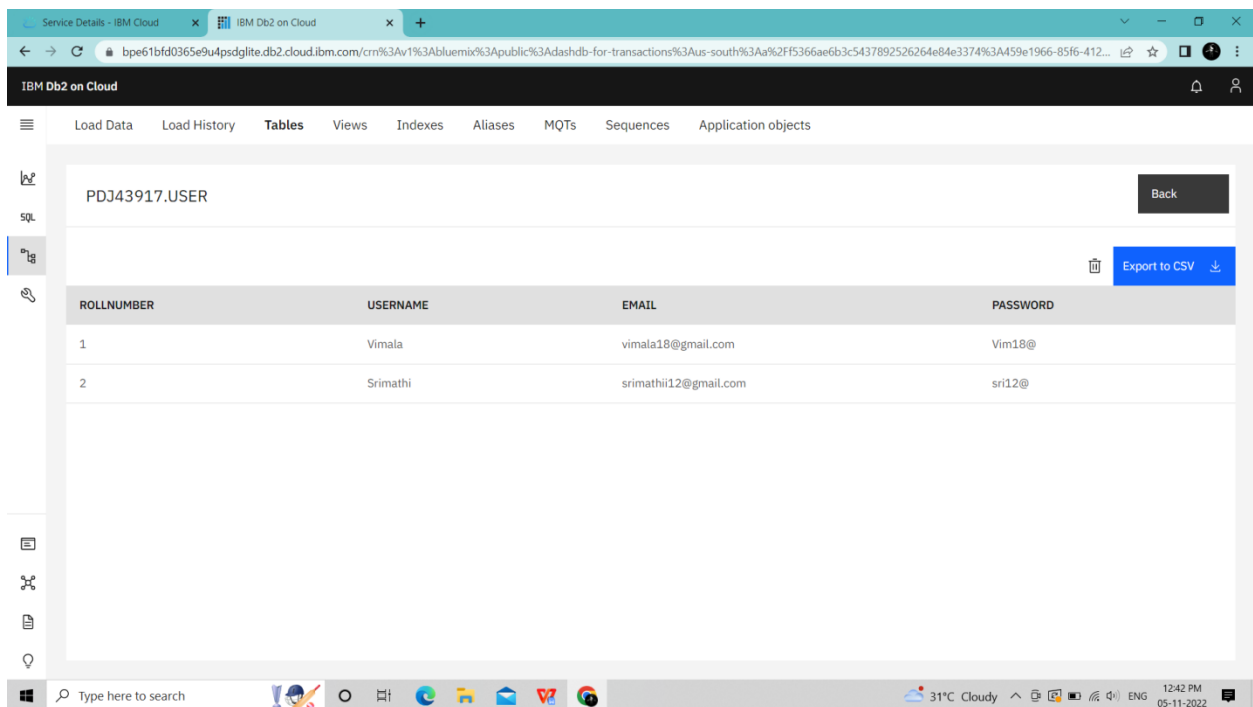


The screenshot shows the IBM Db2 on Cloud SQL editor interface. The left sidebar displays the 'Data objects' tab with a search filter and a list of objects, including 'PDJ43917'. The main editor area shows a query titled '*Untitled - 1' with the following SQL statement:

```
UPDATE USER SET "PASSWORD"='Vim18@'WHERE ROLLNUMBER = '1';
```

Below the query editor, the 'History' tab is active, showing a table of executed queries:

Script	Date	Status	Runtime
Untitled - 1	Nov 5, 2022 12:41:38 PM	✓ 1	0.005 s
UPDATE USER SET "PASSWORD"='Vim18@' WHERE ROLLNUMBER = '1'		✓	0.005 s
Untitled - 1	Nov 5, 2022 12:48:49 PM	✓ 1	0.008 s
Untitled - 1	Nov 5, 2022 12:35:17 PM	✓ 2	0.012 s



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

ROLLNUMBER	USERNAME	EMAIL	PASSWORD
1	Vimala	vimala18@gmail.com	Vim18@
2	Srimathi	srimathii12@gmail.com	sri12@

Thus, for ROLL NUMBER 1 the PASSWORD has been updated from "vimala11@" to "Vim18@".

DELETE TABLE :

The screenshot shows the IBM Db2 on Cloud SQL editor interface. The left sidebar displays the 'Data objects' tab with a search filter 'PDJ43917'. The main editor area shows a script titled '*Untitled - 1' with the following SQL statement:

```
1 DELETE FROM USER WHERE ROLLNUMBER = '2';
```

The 'History' tab is open, showing a table of executed scripts:

Script	Date	Status	Runtime
Untitled - 1	Nov 5, 2022 12:43:31 PM	✓ 1	0.006 s
DELETE FROM USER WHERE ROLLNUMBER = '2'		✓	0.006 s
Untitled - 1	Nov 5, 2022 12:41:38 PM	✓ 1	0.005 s
Untitled - 1	Nov 5, 2022 12:40:49 PM	✓ 1	0.008 s
Untitled - 1	Nov 5, 2022 12:35:17 PM	✓ 2	0.012 s

The screenshot shows the IBM Db2 on Cloud interface displaying the details of the 'PDJ43917.USER' table. The table structure is as follows:

ROLLNUMBER	USERNAME	EMAIL	PASSWORD
1	Vimala	vimala18@gmail.com	Vim18@

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 = "bpv31478" # e.g. "abc12345"
6 dsn_pwd = "zKP9mhh8R2ufS8gT" # e.g. "7dRZ3hdt5XN6$00"
7 dsn_driver = "[IBMDB2CL1]"
8 dsn_database = "bludb"
9 dsn_port = "72304" # e.g. "5000"
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

Username

Shayu

Phone Number

9894222470

Password

.....

Confirm Password

.....

Register

Login Page:-

Login

Phone Number

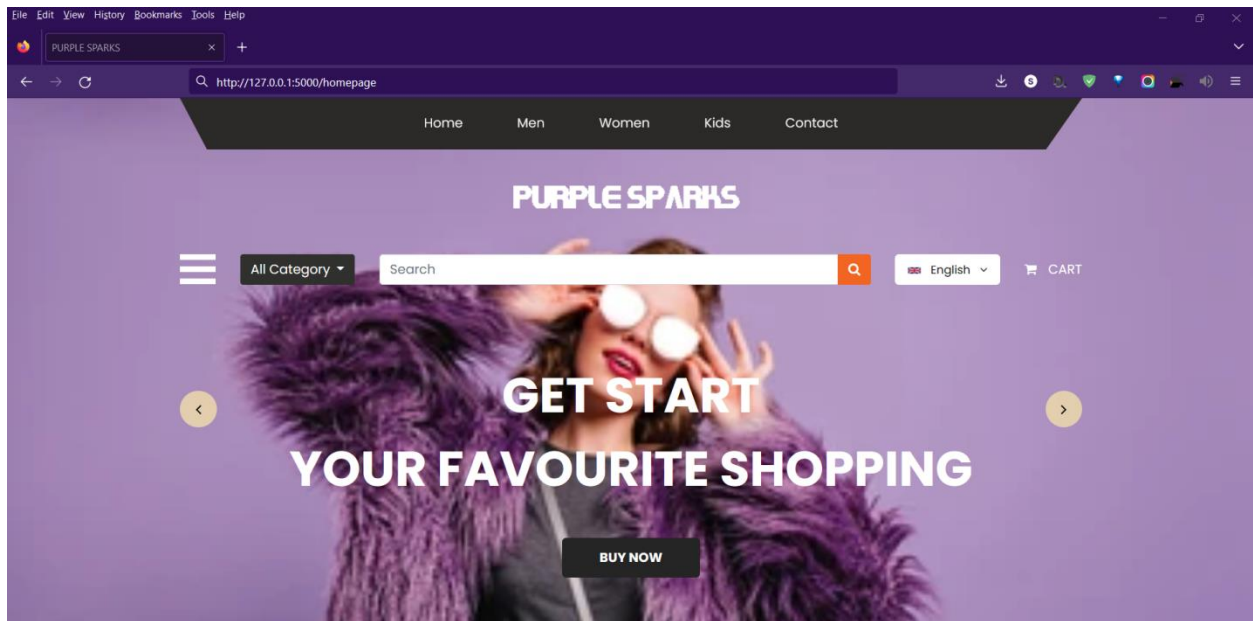
9894222470

Password

.....

Login

Welcome page:-



Database Table :-

A screenshot of the IBM Db2 on Cloud console. The browser address bar shows a URL starting with 'https://bpe61bfd0365e9u4psdglite.db2.cloud.ibm.com/'. The console interface includes a top navigation bar with options like 'Load Data', 'Load History', 'Tables', 'Views', 'Indexes', 'Aliases', 'MQTs', 'Sequences', and 'Application objects'. The 'Tables' tab is selected. On the left sidebar, there are icons for 'SQL' and 'Table'. The main area displays a table named 'BPV31478.USER'. Above the table, there is a 'Back' button. To the right of the table, there is an 'Export to CSV' button with a download icon. The table itself has four columns: 'USERNAME', 'PHONE', 'PASSWORD', and 'CPASSWORD'. It contains one data row with the following values: 'Shayu', '9894222470', 'jmjk7', and 'jmjk7'.

USERNAME	PHONE	PASSWORD	CPASSWORD
Shayu	9894222470	jmjk7	jmjk7