

## Assignment-2

Assignment Date	29 September 2022
Name	R.Karthikeyan
Project Name	CUSTOMER CARE REGISTRY
Maximum mark	2 Mark

### Questions: 1.

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

The screenshot displays the IBM Db2 on Cloud web interface. The top navigation bar includes options like 'Load Data', 'Load History', 'Tables', 'Views', 'Indexes', 'Aliases', 'MQTs', 'Sequences', and 'Application objects'. The 'Tables' section is active, showing a list of tables with columns 'Name', 'Schema', and 'Properties'. A table named 'USERTABLE' is listed under schema 'QNB92291'. The 'Table definition' panel on the right shows the structure of 'USERTABLE' with the following columns:

Name	Data type	Nullable	Length	Scale
USERNA ME	CHAR	Y	5	0
ROLL NUMBER	INTEGER	Y		0
EMAIL	CHAR	Y	5	0
PASSWO				

The interface also includes a search bar for schemas or tables, a 'New table' button, and a 'View data' button. The bottom status bar shows the system temperature as 29°C Cloudy and the time as 09:55 AM on 21-10-2022.

IBM Db2 on Cloud

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

QNB92291.USERTABLE Back

USERNAME	ROLL NUMBER	EMAIL	PASSWORD
----------	-------------	-------	----------

There is no data here yet

29°C Cloudy 09:57 AM 21-10-2022

2. Perform UPDATE,DELETE Queries with user table.

IBM Db2 on Cloud

Data objects My script

Filter objects

QNB92291

\*Untitled - 1

```
1 INSERT INTO USERTABLE VALUES('ARI',4005,'@mail','0987654')
```

History

Script	Date	Status	Runtime
Untitled - 1	Oct 21, 2022 10:12:00 AM	1	0.008 s
INSERT INTO USERTABLE VALUES('ARI',4005,'@mail','0987654')			0.008 s

10:12 AM 21-10-2022

IBM Db2 on Cloud

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

QNB92291.USERTABLE

Back

Export to CSV

USERNAME	ROLL NUMBER	EMAIL	PASSWORD
ARI	4005	.com	0987654
ARI	4005	@mail	0987654

29°C Cloudy 10:13 AM 21-10-2022

## DELETE TABLE:

IBM Db2 on Cloud

Data objects My script

Filter objects

QNB92291

\*Untitled - 1

Beta Classic

Syntax assistant Run all

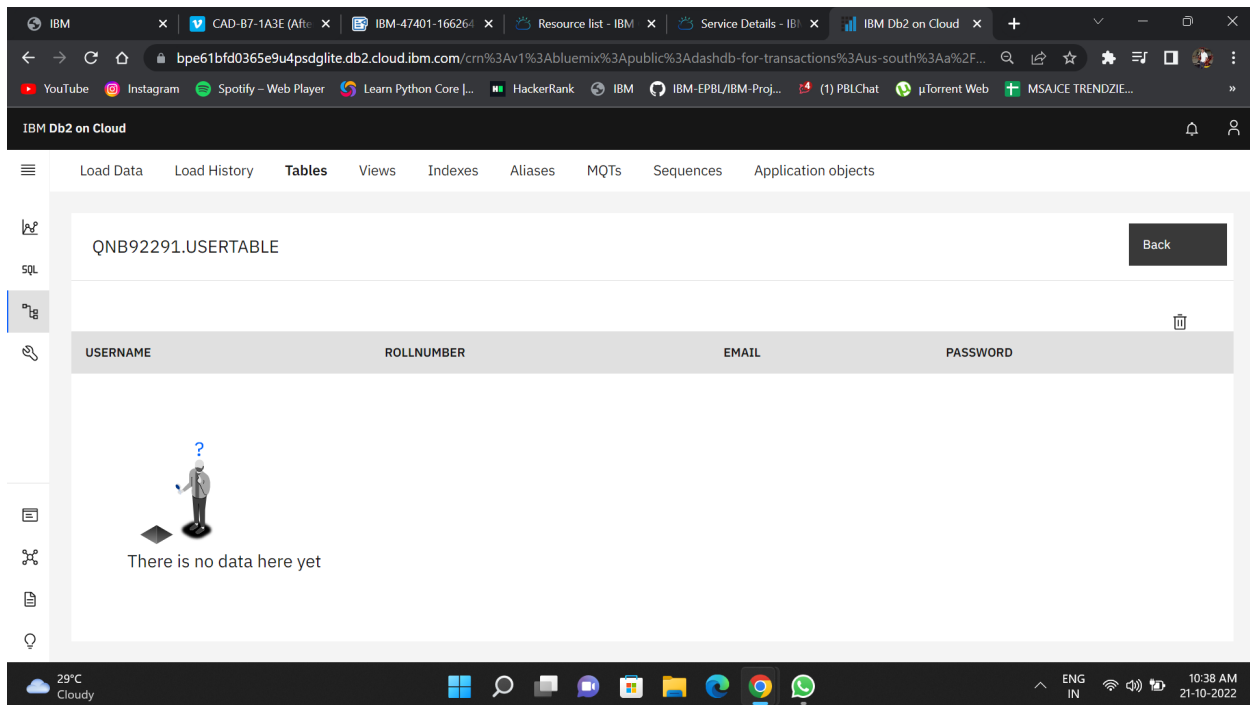
```
1 DELETE FROM USERTABLE WHERE ROLLNUMBER = 4005;
```

History Results

Find history

Script	Date	Status	Runtime
Untitled - 1	Oct 21, 2022 10:24:14 AM	✓ 1	0.010 s
DELETE FROM USERTABLE WHERE ROLLNUMBER = 4005		✓	0.010 s
Untitled - 1	Oct 21, 2022 10:24:03 AM	✗ 1	0.012 s
DELETE FROM USERTABLE WHERE ROLLNO = 4005		✗	0.012 s
Untitled - 1	Oct 21, 2022 10:22:07 AM	✓ 1	0.009 s

29°C Cloudy 10:24 AM 21-10-2022



### 3.Connect python code to db2

```

1 from flask import Flask, render_template, request, redirect,
  url_for, session
2 import ibm_db
3 import re
4 app = Flask(__name__)
5 app.secret_key = 'a'
6 conn =
  ibm_db.connect("DATABASE=User;HOSTNAME=bpe61bfd0365e9u4psdglite
  1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=5671;USERNAME=
  Ari9767;PASSWORD=89gjhayugytueygg;SECURITY=SSL;
7 tGlobalRootCA.crt;", "", "")
8 @app.route("/", methods=['GET', 'POST'])
9 def register():
10 msg = ''
11 if request.method == 'POST':
12 username = request.form['username']
13 email = request.form['email']
14 password = request.form['password']
15 sql = "SELECT * FROM users WHERE username =?"
16 stmt = ibm_db.prepare(conn, sql)
17 ibm_db.bind_param(stmt, 1, username)

```

```
18 ibm_db.execute(stmt)
19 account = ibm_db.fetch_assoc(stmt)
20 print(account)
21 if account:
22     msg = 'Account already exists !'
23 elif not re.match(r'^@+@[^@]+\.[^@]+', email):
24     msg = 'Invalid email address !'
25 elif not re.match(r'[A-Za-z0-9]+', username):
26     msg = 'name must contain only characters and numbers !'
27 else:
28     insert_sql = "INSERT INTO users VALUES (?, ?, ?)"
29     prep_stmt = ibm_db.prepare(conn, insert_sql)
30     ibm_db.bind_param(prepare_stmt, 1, username)
31     ibm_db.bind_param(prepare_stmt, 2, email)
32     ibm_db.bind_param(prepare_stmt, 3, password)
33     ibm_db.execute(prepare_stmt)
34     msg = 'You have successfully registered !'
35 elif request.method == 'POST':
36     msg = 'Please fill out the form !'
37     return render_template('register.html', msg=msg)
38 @app.route('/login', methods=['GET', 'POST'])
39 def login():
40     global userid
41     msg = ''
42     if request.method == 'POST':
43         username = request.form['username']
44         password = request.form['password']
45         sql = "SELECT * FROM users WHERE username =? AND password=?"
46         stmt = ibm_db.prepare(conn, sql)
47         ibm_db.bind_param(stmt, 1, username)
48         ibm_db.bind_param(stmt, 2, password)
49         ibm_db.execute(stmt)
50         account = ibm_db.fetch_assoc(stmt)
51         print(account)
52         if account:
53             session['loggedin'] = True
54             session['id'] = account['USERNAME']
55             userid = account['USERNAME']
56             session['username'] = account['USERNAME']
57             msg = 'Logged in successfully !'
```

```

58 msg = 'Logged in successfully !'
59 return render_template('dashboard.html', msg=msg)
60 else:
61 msg = 'Incorrect username / password !'
62 return render_template('login.html', msg=msg)
63 if __name__ == '__main__':
64 app.run(host='0.0.0.0')
65 # app.run(debug=True)

```

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, session
2 import ibm_db
3 import re
4 app = Flask(__name__)
5 app.secret_key = 'a'
6 conn =
  ibm_db.connect("DATABASE=User;HOSTNAME=bpe61bfd0365e9u4psdglite.c
  logj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=5671;USERNAME=
  Ari9767;PASSWORD=89gjhayugytueygg;SECURITY=SSL;
7 tGlobalRootCA.crt;", "", "")
8 @app.route("/", methods=['GET', 'POST'])
9 def register():
10 msg = ''
11 if request.method == 'POST':
12 username = request.form['username']
13 email = request.form['email']
14 password = request.form['password']
15 sql = "SELECT * FROM users WHERE username =?"
16 stmt = ibm_db.prepare(conn, sql)
17 ibm_db.bind_param(stmt, 1, username)
18 ibm_db.execute(stmt)
19 account = ibm_db.fetch_assoc(stmt)

```

```
20 print(account)
21 if account:
22 msg = 'Account already exists !'
23 elif not re.match(r'^@+@[^@]+\.[^@]+', email):
24 msg = 'Invalid email address !'
25 elif not re.match(r'[A-Za-z0-9]+', username):
26 msg = 'name must contain only characters and numbers !'
27 else:
28 insert_sql = "INSERT INTO users VALUES (?, ?, ?)"
29 prep_stmt = ibm_db.prepare(conn, insert_sql)
30 ibm_db.bind_param(prepare_stmt, 1, username)
31 ibm_db.bind_param(prepare_stmt, 2, email)
32 ibm_db.bind_param(prepare_stmt, 3, password)
33 ibm_db.execute(prepare_stmt)
34 msg = 'You have successfully registered !'
35 elif request.method == 'POST':
36 msg = 'Please fill out the form !'
37 return render_template('register.html', msg=msg)
38 @app.route('/login', methods=['GET', 'POST'])
39 def login():
40 global userid
41 msg = ''
42 if request.method == 'POST':
43 username = request.form['username']
44 password = request.form['password']
45 sql = "SELECT * FROM users WHERE username =? AND password=?"
46 stmt = ibm_db.prepare(conn, sql)
47 ibm_db.bind_param(stmt, 1, username)
48 ibm_db.bind_param(stmt, 2, password)
49 ibm_db.execute(stmt)
50 account = ibm_db.fetch_assoc(stmt)
51 print(account)
52 if account:
53 session['loggedin'] = True
54 session['id'] = account['USERNAME']
55 userid = account['USERNAME']
56 session['username'] = account['USERNAME']
57 msg = 'Logged in successfully !'
58 msg = 'Logged in successfully !'
59 return render_template('dashboard.html', msg=msg)
```

```
60 else:
61 msg = 'Incorrect username / password !'
62 return render_template('login.html', msg=msg)
63 if __name__ == '__main__':
64 app.run(host='0.0.0.0')
65 # app.run(debug=True)
```

## Login:

```
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 <title>Customer Care Login</title>
8 <!-- favicon -->
9 <!-- <link rel="shortcut icon" href="/assets/img/favicon.ico"
10 type="image/x-icon"> -->
11 <!-- <link rel="icon" href="/assets/img/favicon.ico"
    type="image/x-icon">
12 -->
13 <link rel="icon" type="image/png" sizes="16x16"
    href="/assets/img/favicon32x32.png">
14 <!-- bootstrap css cdn -->
15 <link rel="stylesheet"
16 href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/boot
    strap.min.css
17 "
18 integrity="sha384-
19 JcKb8q3iqJ61gNV9KGb8thSsNjpsL0n8PARn9HuZOnIxN0hoP+VmmDGMN5t9UJ0Z"
20 crossorigin="anonymous">
21 <link rel="stylesheet"
    href="https://cdn.jsdelivr.net/npm/font-awesome@4.7.0/css/
    font-awesome.css">
22 <!-- css stylesheet -->
23 <link rel="stylesheet" href="/static/css/style2.css">
24 <!-- font styles cdn -->
```



```

25 <link rel="preconnect" href="https://fonts.gstatic.com">
26 <link
27 href="https://fonts.googleapis.com/css2?family=Alegreya&display=s
wap"
28 rel="stylesheet">
29 <link
30 href="https://fonts.googleapis.com/css2?family=Alegreya:wght@600&
display=swap"
31 rel="stylesheet">
32 </head>
33 <body>
34 <div class="login text-center mt-5">
35 <form action="/login" method="post">
36 <h2 class="form-text-h2"> Login Form </h2>
37 <div class="msg">{{ msg }}</div>
38 <div class="form-group mb-3">
39 <label class="label" for="username">Username</label>
40 <input type="text" class="form-control" placeholder="Username"
41 name="username" required>
42 </div>
43 <div class="form-group mb-3">
44 <label class="label" for="password">Password</label>
45 <input type="password" class="form-control"
46 placeholder="Password" name="password" required>
47 </div>
48 <button type="submit" id="button" class="btn btn-primary"> Login
49 </button>
50 <div class="note mt-3 text-center">
51 <!--Register form -->
52 <p> Don't have an account yet? Click here to <a
53 href="/">register </a> </p>
54 </div>
55 </form>
56 </div>
57 </body>
58 </html>

```

**Register:**

```
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 <title> Customer care Registry</title>
8 <!-- bootstrap css cdn -->
9 <link rel="stylesheet"
10 href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstr
  ap.min.css
11 "
12 integrity="sha384-
13 JcKb8q3iqJ61gNV9KGb8thSsNjpSL0n8PARn9HuZOnIxN0hoP+VmmDGMN5t9UJ0Z"
14 crossorigin="anonymous">
15 <link rel="stylesheet"
  href="https://cdnjs.cloudflare.com/ajax/libs/fontawesome/4.7.0/css/f
  ont-awesome.css">
16 <!-- css stylesheet -->
17 <link rel="stylesheet" href="/static/css/style.css">
18 <!-- font styles cdn -->
19 <link rel="preconnect" href="https://fonts.gstatic.com">
20 <link
21 href="https://fonts.googleapis.com/css2?family=Alegreya&display=swa
  p"
22 rel="stylesheet">
23 <link
24 href="https://fonts.googleapis.com/css2?family=Alegreya:wght@600&dis
  play=swap"
25 rel="stylesheet">
26 </head>
27 <body>
28 <!-- bootstrap navbar -->
```

```

29 <!-- navbar ends -->
30 <!-- Login form -->
31 <div class="login text-center mt-5">
32 <form action="/" method="post">
33 <h2 class="form-text-h2"> Register Form </h2>
34 <div class="msg">{{ msg }}</div>
35 <input type="text" class="form-control" name="username"
36 placeholder="Enter Your Username" id="username"
37 required>
38 <input type="email" class="form-control" name="email"
39 placeholder="Enter Your Email ID" id="email" required>
40 <input type="password" class="form-control" name="password"
41 placeholder="Enter Your Password" id="password"
42 required>
43 <button type="submit" id="button" class="btn btn-primary">
44 Register </button>
45 <div class="note mt-3 text-center">
46 <!--Register form -->
47 <p> already have an account ? please login <a
48 href="/login">login.. </a> </p>
49 </div>
50 </form>
51 </div>
52 </body>
53 </html>

```

### Style.css

```

1 * {
2     margin: 0;
3     padding: 0;
4 }
5 body {
6     background-color: aqua !important;

```

```
7     display: flex;
8     flex-direction: column;
9     align-items: center;
10    justify-content: center;
11    }
12    form {
13    display: flex;
14    flex-direction: column;
15    align-items: center;
16    justify-content: center;
17    background-color: white;
18    width: 40%;
19    height: 450px;
20    border-radius: 50px;
21    margin: 100px;
22    gap: 25px;
23    }
24    h2 {
25    font-size: large;
26    font-weight: bolder;
27    }
28    input {
29    width: 80% !important;
30    }
31    div {
32    display: flex;
33    align-items: center;
34    justify-content: center;
35    flex-direction: column;
36    width: 90%;
37    }
38    .form-text-vk {
39    font-size: small;
40    padding-bottom: 10px;
41    }
42    .form-text-h2 {
43    font-size: x-large;
44    padding-top: 10px;
45    }
```

The DataBase is created:

The screenshot displays the IBM Db2 on Cloud web interface. The browser address bar shows the URL: `bpe61bfd0365e9u4psdglite.db2.cloud.ibm.com/cm%3Av1%3Abluemix%3Apublic%3Adashdb-for-transactions%3Aus-south%3Aa%2Fe6b0...`. The page title is "IBM Db2 on Cloud".

The main content area shows a table named "QNB92291.USER". The table has three columns: "USERNAME", "EMAIL", and "PASSWORD". The table contains one row of data:

USERNAME	EMAIL	PASSWORD
Ari	tariharn6@gmail.com	12345

Navigation and Action Elements:

- Top Navigation Bar:** Includes tabs for "Load Data", "Load History", "Tables", "Views", "Indexes", "Aliases", "MQTs", "Sequences", and "Application objects".
- Left Sidebar:** Contains icons for "SQL", "Tables", "Views", and "Indexes".
- Table Actions:** A "Back" button is located in the top right corner of the table view. An "Export to CSV" button with a download icon is located in the bottom right corner of the table view.
- Bottom Taskbar:** Shows system information: "32°C Cloudy", taskbar icons, and the time "11:32 AM 21-10-2022".