

Assignment-2

Assignment Date	29 September 2022
Name	Navaneethan P
Project Name	Smart Fashion Recommender Application
Maximum Marks	2 mark

Questions 1:

Create User table with user with email,username,rollnumber,password

The screenshot shows the IBM Db2 on Cloud console interface. The top navigation bar includes options like Load Data, Load History, Tables, Views, Indexes, Aliases, MQTs, Sequences, and Application objects. The 'Tables' tab is selected. On the left, a search bar is present. The main area displays a list of tables under the 'Tables' section. A table named 'USERTABLE' is listed with schema 'VVB28404'. To the right, the 'Table definition' panel shows the table's structure:

Name	Data type	Nullable	Length	Scale
NAME	CHAR	Y	10	0
ROLLNUMBER	INTEGER	Y		0
MAIL	VARCHAR	Y	32	0
PASSWORD	VARCHAR	Y	32	0

Below the table definition, there is a 'View data' button. The bottom status bar shows the Windows taskbar with the search bar and system clock indicating 2:30 PM on 2022-10-27.

IBM Db2 on Cloud

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

VVB28404.USERTABLE Back

Export to CSV

NAME	ROLLNUMBER	MAIL	PASSWORD
Alexa	1	alexa@gmail.com	12345

Type here to search

2:33 PM 2022-10-27

2.Perform UPDATE,DELETE Queries with user table.

Service Details - IBM IBM Db2 on Cloud Your Platform tempor IBM GitHub - IBM-EPBL/IB Code Search

bpe61bfd0365e9u4psdglite.db2.cloud.ibm.com/crn%3Av1%3Abluemix%3Apublic%3Adashdb-for-transactions%3Aus-south%3Aa%2Fcea66a5e82804ec0a15b...

IBM Db2 on Cloud

Data objects My script

Filter objects

VVB28404

*Untitled - 1

Beta Classic

```
1 INSERT INTO USERTABLE VALUES ('ALEXA','311819104001','alexa1029@gmail.com','9201axela');
2
3
4
5
```

Run all

History

Type here to search

11:36 AM 2022-10-27

Service Details - IBM Db2 on Cloud

IBM Db2 on Cloud

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

VVB28404.USERTABLE

Back

Export to CSV

USERNAME	REG No	MAIL ID	PASSWORD
ALEXA	1	alexa1029@gmail.com	9201axela

Type here to search

11:40 AM 2022-10-27

DELETE TABLE:

Service Details - IBM Db2 on Cloud

IBM Db2 on Cloud

Data objects My script

Filter objects

VVB28404

*Untitled - 1

Beta Classic

Syntax assistant

Run all

```

1 delete from USERTABLE where ROLLNUMBER = 01;
2
3
4

```

History

Find history

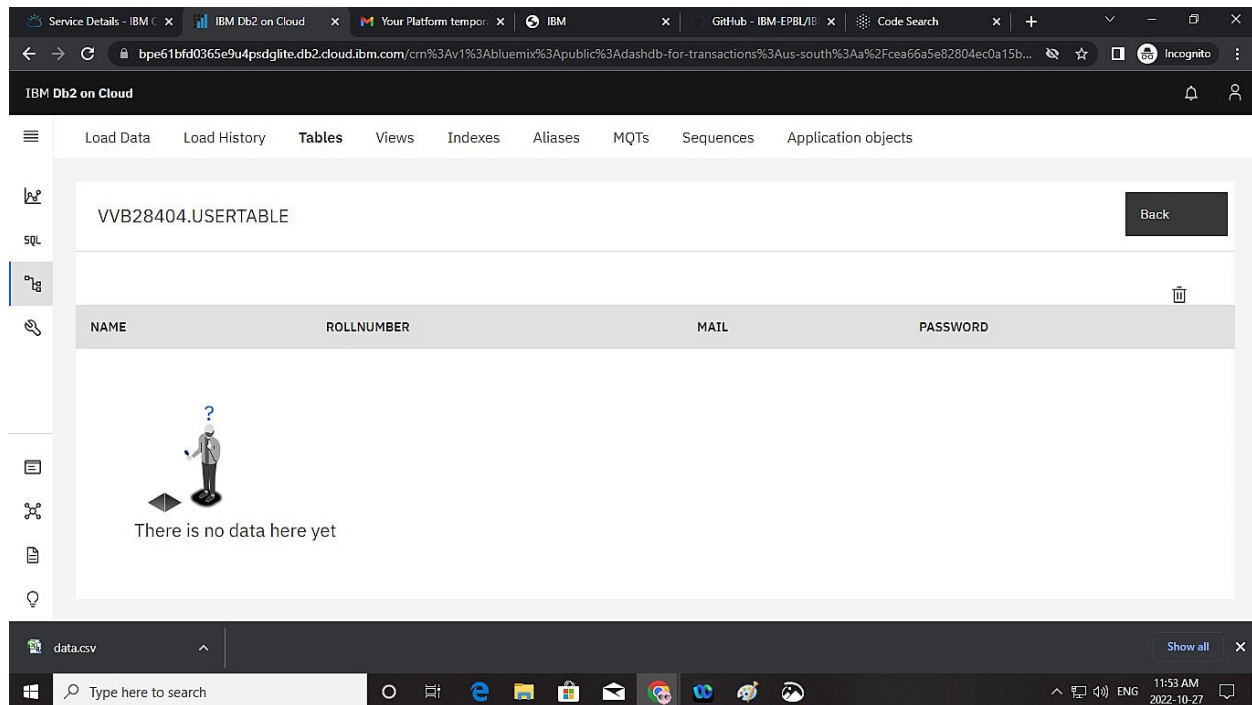
Script	Date	Status	Runtime
Untitled - 1	Oct 27, 2022 11:52:48 AM	1	0.007 s
delete from USERTABLE where ROLLNUMBER = 01			0.007 s
Untitled - 1	Oct 27, 2022 11:51:33 AM	1	0.006 s
INSERT INTO USERTABLE VALUES ('ALEXA','01','example@','09867')			0.006 s
Untitled - 1	Oct 27, 2022 11:47:15 AM	1	0.019 s

data.csv

Show all

Type here to search

11:52 AM 2022-10-27



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=b70af05b-76e4-4bca-
   a1f5-
   23dbb4c6a74e.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=
   5671;USERNAME=<user name>;PASSWORD=*****;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 msg = ''
41 if request.method == 'POST':
42 username = request.form['username']
43 password = request.form['password']
44 sql = "SELECT * FROM users WHERE username =? AND password=?"
45 stmt = ibm_db.prepare(conn, sql)
46 ibm_db.bind_param(stmt, 1, username)
47 ibm_db.bind_param(stmt, 2, password)
48 ibm_db.execute(stmt)
49 account = ibm_db.fetch_assoc(stmt)
50 print(account)
51 if account:
52 session['loggedin'] = True
53 session['id'] = account['USERNAME']
54 userid = account['USERNAME']
55 session['username'] = account['USERNAME']
56 msg = 'Logged in successfully !'

```

```

57 msg = 'Logged in successfully !'
58 return render_template('dashboard.html', msg=msg)
59 else:
60 msg = 'Incorrect username / password !'
61 return render_template('login.html', msg=msg)
62 if __name__ == '__main__':
63 app.run(host='0.0.0.0')
64 # 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=b70af05b-76e4-4bca-
  a1f5-
  23dbb4c6a74e.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=
  5671;USERNAME=<user name>;PASSWORD=*****;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,
  initialscales=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 <link rel="icon" type="image/png"
    sizes="16x16" href="/assets/img/favicon32x32.png">
13 <!-- bootstrap css cdn -->
14 <link rel="stylesheet"
15 href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/boot
  strap.min.css"
16 integrity="sha384JcKb8q3iqJ61gNV9KGb8thSsNjpSL0n8PARn9HuZ0nIxN0ho
  P+VmmDGMN5t9UJ0Z" crossorigin="anonymous"><link
    rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font
    awesome/4.7.0/css/font awesome.css">
17 <!-- css stylesheet -->
18 <link rel="stylesheet" href="/static/css/style2.css">
19 <!-- font styles cdn -->
20 <link rel="preconnect" href="https://fonts.gstatic.com">
21 <link
22 href="https://fonts.googleapis.com/css2?family=Alegreya&display=s
  wap"
```



```

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

```

```
    initials=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/boot
    strap.min.css"
11 integrity="sha384JcKb8q3iqJ61gNV9KGb8thSsNjpSL0n8PARn9HuZ0nIxN0ho
    P+VmmDGMN5t9UJ0Z"crossorigin="anonymous">
12 <linkrel="stylesheet"href="https://cdnjs.cloudflare.com/ajax/libs
    /fontawesome/4.7.0/css/font-awesome.css">
13 <!-- css stylesheet -->
14 <link rel="stylesheet" href="/static/css/style.css">
15 <!-- font styles cdn -->
16 <link rel="preconnect" href="https://fonts.gstatic.com">
17 <link
18 href="https://fonts.googleapis.com/css2?family=Alegreya&display=s
    wap"
19 rel="stylesheet">
20 <link
    href="https://fonts.googleapis.com/css2?family=Alegreya:wght@600&
    display=swap"
21 rel="stylesheet">
22 </head>
23 <body>
24 <!-- bootstrap navbar -->
25 <!-- navbar ends -->
26 <!-- Login form -->
27 <div class="login text-center mt-5">
28 <form action="/" method="post">
29 <h2 class="form-text-h2"> Register Form </h2>
30 <div class="msg">{{ msg }}</div>
31 <input type="text" class="form-control" name="username"
32 placeholder="Enter Your Username" id="username"
33 required>
34 <input type="email" class="form-control" name="email"
35 placeholder="Enter Your Email ID" id="email" required>
36 <input type="password" class="form-control" name="password"
37 placeholder="Enter Your Password" id="password"
38 required>
39 <button type="submit" id="button" class="btn btn-primary">
```

```
40 Register </button>
41 <div class="note mt-3 text-center">
42 <!--Register form -->
43 <p> already have an account ? please login <a
44 href="/login">login.. </a> </p>
45 </div>
46 </form>
47 </div>
48 </body>
49 </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:

Sent Mail - navaneet: xGitHub - IBM-EPBL/IB xIBM xIBM-42802-1662634 xService Details - IBM xIBM Db2 on Cloud x

bpe61bfd0365e9u4psdglite.db2.cloud.ibm.com/cn%3Av1%3Abluemix%3Apublic%3Adashdb-for-transactions%3Aus-south%3Aa%2Fcea66a5e82804ec0a15b...

IBM Db2 on Cloud

Load DataLoad HistoryTablesViewsIndexesAliasesMQTsSequencesApplication objects

VVB28404.USERTABLEBack

Export to CSV

NAME	ROLLNUMBER	MAIL	PASSWORD
Alexa	1	alexa@gmail.com	12345

Type here to search

3:14 PM2022-10-27