

Assignment -2

IBM CLOUD DB2

Assignment Date	1 OCTOBER 2022
Student Name	GOWSEELAN K
Student Roll Number	812019205012
Maximum Marks	2 Marks

Questions:

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

The screenshot displays the IBM Db2 on Cloud console interface. At the top, there's a navigation bar with tabs for GitHub, IBM-EPBL/IBM-Project-43742-16, Service Details - IBM Cloud, and IBM Db2 on Cloud. Below this is a search bar and a list of bookmarks. The main content area is titled 'IBM Db2 on Cloud' and features a red error banner that reads: 'Error: Please check network connectivity then try again.' Below the error banner, there's a search bar for 'Find schemas or tables' and a 'Refresh' button. The left sidebar contains icons for SQL, Schemas, and other database management tools. The main panel is divided into two sections: 'Tables' and 'Table definition'. The 'Tables' section shows a list of tables with columns 'Name', 'Schema', and 'Properties'. A table named 'CUSTOMER_DATA' is listed under the 'FLK26349' schema. The 'Table definition' section shows the structure of the 'CUSTOMER_DATA' table with columns: USERNAME (CHAR, 32, NULLABLE), ROLLNUMBER (INTEGER, NULLABLE), EMAILID (VARCHAR, 32, NULLABLE), and PASSWORD (VARCHAR, 32, NULLABLE). A 'View data' button is visible at the bottom of the table definition section. The bottom of the screenshot shows the Windows taskbar with the search bar, task view button, and several open applications, along with the system clock showing 6:31 PM on 11/8/2022.

IBM Db2 on Cloud

Error: Please check network connectivity then try again. [Show logs](#)

Find schemas or tables [Refresh](#)

Tables [New table](#)

Name	Schema	Properties
CUSTOMER_DATA	FLK26349	...

Total: 1, selected: 0

Table definition [CUSTOMER_DATA](#) No statistics available.

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

[View data](#)

- Perform UPDATE,DELETE Queries with user table.

UPDATE TABLE:-

The screenshot displays the IBM Db2 on Cloud web interface. At the top, a browser window shows the URL: `https://bpe61bfd0365e9u4psdglite.db2.cloud.ibm.com/cm%3Av1%3Abluemix%3Apublic%3Adashdb-for-transactions%3Aus-south%3Aa%2F97fc099ed0194c76963a1221c8339f04%3Ac7d903...`. The interface includes a sidebar with navigation icons for SQL, Tables, Views, and Functions. The main area features a red error banner at the top: "Error: Please check network connectivity then try again." Below this, a SQL editor shows a single query: `INSERT INTO Customer_data values ('rahu1','30','ragul.it19@mamcet.com','rahu101');`. The "Run all" button is visible. A "History" tab is active, displaying a table of executed scripts.

Script	Date	Status	Runtime
Untitled - 1	Nov 8, 2022 6:38:55 PM	✓ 1	0.006 s
INSERT INTO Customer_data values ('rahu1','30','ragul.it19@mamcet.com','rahu...		✓	0.006 s

The Windows taskbar at the bottom shows the system clock as 6:39 PM on 11/8/2022, with a temperature of 26°C and a "Cloudy" weather status.

GitHub

IBM-EPBL/IBM-Project-43742-16

Service Details - IBM Cloud

IBM Db2 on Cloud

https://bpe61bfd0365e9u4psdglite.db2.cloud.ibm.com/crn%3Av1%3Abluemix%3Apublic%3Adashdb-for-transactions%3Aus-south%3Aa%2F97fc099ed0194c76963a1221c8339f04%3Ac7d903...

PersonalBrittany ChiangGmailYouTubeMapsQ!Manage Payment...

Other bookmarks

IBM Db2 on Cloud

Load DataLoad HistoryTablesViewsIndexesAliasesMQTsSequencesApplication objects

FLK26349.CUSTOMER_DATA

Back

Export to CSV

Type here to search

AQI 65.78ENG IN6:44 PM11/8/2022

DELETE TABLE:-

GitHub

IBM-EPBL/IBM-Project-43742-16

Service Details - IBM Cloud

IBM Db2 on Cloud

https://bpe61bfd0365e9u4psdglite.db2.cloud.ibm.com/crm%3Av1%3Abluemix%3Apublic%3Adashdb-for-transactions%3Aus-south%3Aa%2F97fc099ed0194c76963a1221c8339f04%3Ac7d903...

PersonalBrittany ChiangGmailYouTubeMapsQ!Manage Payment...

Other bookmarks

IBM Db2 on Cloud

Data objects

Saved objects

Filter objects

FLK26349

*Untitled - 1

SQL

DEDELETE FROM CUSTOMER_DATA WHERE USERNAME = 'nithish';

History

Find history

Script	Date	Status	Runtime
Untitled - 1	Nov 8, 2022 7:00:38 PM	1	0.006 s
DEDELETE FROM CUSTOMER_DATA WHERE USERNAME = 'nithish'			0.006 s

Type here to search

29°C Mostly cloudy

ENG IN

7:01 PM

11/8/2022

FLK26349.CUSTOMER_DATA

Back

Export to CSV

USERNAME	ROLLNUMBER	EMAILID	PASSWORD
rahul	30	ragul.it19@mamcet.com	rahul01

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

Answers

App.py

```
from flask import Flask, render_template, request,
redirect, url_for, session import ibm_db import re app =
Flask(__name__) app.secret_key =
'a' conn
=
ibm_db.connect(
    "DATABASE=Db2New;HOSTNAME=19af6446-6171-4641-8aba-
9dcff8e1b6ff.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=30699;USERNAM
E=kgb92696;PASSWORD=3IlWyGYU3bMIvwnX;SECURITY=SSL;SSLSERVERCERTIFICATE=DigiCer
tGlobalRootCA.crt;", "", "")

@app.route("/", methods=['GET', 'POST'])
def register():    msg = ''    if
request.method == 'POST':
    username = request.form['username']
email = request.form['email']    password =
request.form['password']    sql = "SELECT * FROM
users 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:
msg = 'Account already exists !'    elif
not re.match(r'^@+@[^@]+\.[^@]+', email):

    msg = 'Invalid email address !'    elif
not re.match(r'[A-Za-z0-9]+', username):
    msg = 'name must contain only characters and numbers !'
```

```

        else:
            insert_sql = "INSERT INTO users VALUES (?, ?, ?)"
            prep_stmt = ibm_db.prepare(conn, insert_sql)
            ibm_db.bind_param(prepare_stmt, 1, username)
            ibm_db.bind_param(prepare_stmt, 2, email)
            ibm_db.bind_param(prepare_stmt, 3, password)
            ibm_db.execute(prepare_stmt)
            msg = 'You have successfully registered !'
            elif request.method == 'POST':
                msg = 'Please fill out the form !'
            return render_template('register.html', msg=msg)

@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 users 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:
            session['loggedin'] = True
            session['id'] = account['USERID']
            session['username'] = account['USERNAME']
            msg = 'Logged in successfully !'
        else:
            msg = 'Incorrect username / password !'
        return render_template('login.html', msg=msg)

if __name__ == '__main__':
    app.run(host='0.0.0.0')
    # app.run(debug=True)

```

Signup.html

```

<!DOCTYPE html>
<head>
<title>signup page</title>
<link href={{url_for('static',filename='signup.css')}} rel="stylesheet">
</head>
<body >

```

```

<div id="form">
<form action="/login" method="post" style="border:1px solid #ccc">
  <div class="container">
    <h1>Sign Up</h1>
    <hr>
    <label for="uname"><b>Username</b></label>
    <input type="text" placeholder="Enter username" name="uname" required>
    <label for="roll"><b>Roll Number</b></label>
    <input type="text" placeholder="Enter roll number" name="roll" required>
    <label for="email"><b>Email</b></label>
    <input type="text" placeholder="Enter Email" name="email" required>
    <label for="psw"><b>Password</b></label>
    <input type="password" placeholder="Enter Password" name="psw" required>

    <label>
      <input type="checkbox" checked="checked" name="remember" style="margin-bottom:15px">
    </label>

    <div class="clearfix">
      <button type="button" class="cancelbtn">Cancel</button>
      <button type="submit" class="signupbtn">Sign Up</button><br><br>
      <p>Already registered user <a href="{{ url_for('signin') }}">signin</a></p>
    </div>
  </div>
</form></div></body>

```

```

<!DOCTYPE html>
<head>
  <title>Signin page</title>
  <link href="{{url_for('static',filename='signup.css')}}" rel="stylesheet">
</head><div id="for">
<form action="action_page.php" method="post" id="forme" >
  <div class="imgcontainer">
  </div>

  <div class="container">
    <h1>Sign in</h1>
    <label for="email"><b>Email</b></label>
    <input type="text" placeholder="Enter email" name="email" required>
    <label for="psw"><b>Password</b></label>

```


Login.html

```
<input type="password" placeholder="Enter Password" name="psw" required>
<button type="submit">sign in</button>
<label>
  <input type="checkbox" checked="checked" name="remember"> Remember me
</label>
</div>
```

```
<div class="container" style="background-color:#f1f1f1">
  <button type="button" class="cancelbtn">Cancel</button>    </div>
</form></div>
```

Style.css

```
* {box-sizing: border-box}

/* Full-width input fields */
input[type=text], input[type=password] {
width: 100%; padding: 15px; margin:
5px 0 22px 0; display: inline-block;
border: none; background: #f1f1f1;
} input[type=text]:focus,
input[type=password]:focus { background-color:
#ddd; outline: none;
} hr { border: 1px solid
#f1f1f1; margin-bottom:
25px;
}

/* Set a style for all buttons
*/ button { background-color:
#04AA6D; color: white;
padding: 14px 20px; margin:
8px 0; border: none; cursor:
pointer; width: 50%;
opacity: 0.9;
```

```

}
button:hover {
opacity:1;
}

/* Extra styles for the cancel button */
.cancelbtn { padding:
14px 20px; background-
color: #f44336;
}

/* Float cancel and signup buttons and add an equal width */
.cancelbtn, .signupbtn
{ float: left;
width: 50%;
}

/* Add padding to container elements */
.container {
padding: 16px;
}

/* Clear floats */
.clearfix::after {
content: ""; clear:
both; display:
table;
}

/* Change styles for cancel button and signup button on extra small screens */
@media screen and (max-width: 300px) {
.cancelbtn, .signupbtn {
width: 50%;
}
}

/* Bordered form */ form {
border: 3px solid #f1f1f1;
}

/* Extra style for the cancel button (red) */ .cancelbtn
{

```



```
    width: 50%;    padding:
10px 18px;    background-
color: #f44336;
}

/* Add padding to containers */
.container {
padding: 16px;
}

/* Change styles for span and cancel button on extra small screens */
@media screen and (max-width: 300px)
{    span.psw {    display: block;
float: none;
}
    .cancelbtn {
width: 50%;
}
} #form{    padding: 25%;
background-color: rgb(172, 223, 223);
} #forme {
padding: 25%;
}
```

OUTPUT:- SIGNUP PAGE:-

Search save form details x Service Details - IBM x IBM Db2 on Cloud x Logs x Getting started with x signup page x

localhost:5000

Sign Up

Username

gan

Roll Number

123

Email

gan@123.com

Password

.....

☒ Remember me

Cancel Sign Up

Already registered user [signin](#)

Type here to search

16:48 05-11-2022

SIGNIN PAGE:-

Search save form details x Service Details - IBM x IBM Db2 on Cloud x Logs x Getting started with x Signin page x

localhost:5000/signin

Sign in

Email

Enter email

Password

Enter Password

sign in ☒ Remember me

Cancel

Type here to search

28°C 16:55 05-11-2022

WELCOME PAGE:-

