

Assignment -2

IBM CLOUD DB2

Assignment Date	29 September 2022
Student Name	Ganesh Kumar
Student Roll Number	812019205009
Maximum Marks	2 Marks

Questions:

- 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 tabs for 'Search save form details in db2', 'Service Details - IBM Cloud', 'IBM Db2 on Cloud', and 'Getting started with Db2 on Cloud'. The main content area is titled 'IBM Db2 on Cloud' and features a sidebar with icons for 'SQL', 'Schemas', and 'Tables'. The 'Tables' tab is active, showing a list of tables under the 'VSV60334' schema. The 'USERDATA' table is selected, and its definition is displayed in a panel on the right. The table definition panel shows the following columns:

Name	Data type	Nullable	Length	Scale
USERNAME	CHAR	Y	5	0
ROLLNUMBER	INTEGER	Y		0
EMAIL	VARCHAR	Y	32	0
PASSWORD	VARCHAR	Y	32	0

Below the table definition, there is a 'View data' button. The bottom status bar indicates 'Total: 1, selected: 0'.

- 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 database named 'VSV60334'. The main editor area shows an SQL statement: `INSERT INTO USERDATA values ('ganes','07','ganeshkumar.it19@mamcet.com','ganesh####');`. Below the editor, the 'History' tab is active, displaying a table of executed scripts.

Script	Date	Status	Runtime
Untitled - 1	Nov 5, 2022 3:32:29 PM	✓ 1	0.007 s
INSERT INTO USERDATA values ('ganes','07','ganeshkumar.it19@mam...		✓	0.007 s

The screenshot shows the IBM Db2 on Cloud interface displaying the contents of the 'VSV60334.USERDATA' table. The table has four columns: USERNAME, ROLLNUMBER, EMAIL, and PASSWORD. The data row shows 'ganes' as the username, '7' as the roll number, 'ganeshkumar.it19@mamcet.com' as the email, and 'ganesh####' as the password. There is an 'Export to CSV' button and a 'Back' button in the top right corner.

USERNAME	ROLLNUMBER	EMAIL	PASSWORD
ganes	7	ganeshkumar.it19@mamcet.com	ganesh####

DELETE TABLE:-

The screenshot shows the IBM Db2 on Cloud console interface. On the left, the 'Data objects' tab is active, displaying a list of objects including 'VSV60334'. The main area shows a SQL script editor with the following code:

```
1 INSERT INTO USERDATA values ('ganes','07','ganeskumar.it19@mamcet.com','ganes@###');
2 INSERT INTO USERDATA values ('alwin','02','alwin.it19@mamcet.com','alwin@###');
3 DELETE FROM USERDATA WHERE USERNAME = 'ganes';
4
```

Below the editor, the 'History' tab is selected, showing a table of execution history:

Script	Date	Status	Runtime
Untitled - 1	Nov 5, 2022 3:38:56 PM	✓ 3	0.023 s
INSERT INTO USERDATA values ('ganes','07','ganeskumar.it19@mam...		✓	0.009 s
INSERT INTO USERDATA values ('alwin','02','alwin.it19@mamcet.co...		✓	0.007 s
DELETE FROM USERDATA WHERE USERNAME = 'ganes'		✓	0.007 s

The screenshot shows the IBM Db2 on Cloud console interface, specifically the 'Tables' tab. The table 'VSV60334.USERDATA' is selected. The table structure is displayed as follows:

USERNAME	ROLLNUMBER	EMAIL	PASSWORD
alwin	2	alwin.it19@mamcet.com	alwin@###

The interface also includes a 'Back' button and an 'Export to CSV' button.

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 !'
            return
        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>

```

Login.html

```

<!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>

```

```

<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 deta x Service Details - IBM x IBM Db2 on Cloud x Logs x Getting started with D 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 deta x Service Details - IBM x IBM Db2 on Cloud x Logs x Getting started with D 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:-

