

ASSIGNMENT 2

Student Name	Anupriya S
Student Roll Number	820519205004
Maximum Marks	2 Marks

Questions:

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

1. Perform UPDATE, DELETE Queries with user table.
2. Connect python code to db2.
3. 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.

Solutions:

1. Creating user table with user email, username, roll number, password.

The screenshot displays the IBM Db2 on Cloud web interface. The browser address bar shows the URL: `bpe61bfd0365e9u4psdglite.db2.cloud.ibm.com/cn%3Av1%3Abluemix%3Apublic%3Adashdb-for-transactions%3Aus-south%3Aa%2F1618829015fa44daa13b06aa2...`. The interface includes a top navigation bar with tabs for 'Service Details - IBM Cloud' and 'IBM Db2 on Cloud'. Below this is a search bar with the text 'Find schemas or tables' and a 'Refresh' button. The main content area is divided into two panels. The left panel, titled 'Tables', shows a list of tables with columns 'Name', 'Schema', and 'Properties'. A table named 'STUDENT' is listed under the 'MCL23949' schema. The right panel, titled 'Table definition', shows the structure of the 'STUDENT' table. It includes a table with the following columns: 'Name', 'Data type', 'Nullable', 'Length', and 'Scale'. The rows are: 'USERNAME' (CHAR, N, 5, 0), 'EMAIL' (VARCHAR, Y, 32, 0), 'Roll number' (VARCHAR, Y, 32, 0), and 'PASSWORD' (VARCHAR, Y, 32, 0). A 'View data' button is located at the bottom of the right panel. The bottom of the screenshot shows a Windows taskbar with various application icons and a system clock indicating 06:43 on 19-10-2022.

Name	Schema	Properties
STUDENT	MCL23949	...

Name	Data type	Nullable	Length	Scale
USERNAME	CHAR	N	5	0
EMAIL	VARCHAR	Y	32	0
Roll number	VARCHAR	Y	32	0
PASSWORD	VARCHAR	Y	32	0

The screenshot displays the IBM Db2 on Cloud web interface. The browser address bar shows the URL: `bpe61bfd0365e9u4psdglite.db2.cloud.ibm.com/cn%3A%3Av1%3A%3Abluemix%3Apublic%3Adashdb-for-transactions%3Aus-south%3Aa%2F1618829015fa44daa13b06aa2...`. The interface is titled "IBM Db2 on Cloud".

On the left, the "Data objects" panel shows a tree structure with the following items: MCL23949, Tables, STUDENT, Views, MQTs, Aliases, and Nicknames. The "My script" tab is active.

The main editor area shows a SQL script in a file named "*Untitled - 1". The script contains two lines of SQL:

```
1 insert into student values('ayish',3,8,1);
2 insert into student values('anu',9,4,6);
3
```

Below the script editor, the "History" panel is visible, showing a table of executed scripts. The table has the following columns: Script, Date, Status, and Runtime.

Script	Date	Status	Runtime
^ Untitled - 1	Oct 19, 2022 6:38:10 AM	✓ 2	0.015 s
insert into student values('ayish',3,8,1)		✓	0.007 s
insert into student values('anu',9,4,6)		✓	0.008 s
v Untitled - 1	Oct 19, 2022 6:37:27 AM	✗ 2	0.022 s
^ Untitled - 1	Oct 19, 2022 6:35:46 AM	✗ 2	0.020 s

The Windows taskbar at the bottom shows the system clock as 06:41 on 19-10-2022.

Output :

The screenshot shows the IBM Db2 on Cloud web interface. The top navigation bar includes tabs for Load Data, Load History, Tables, Views, Indexes, Aliases, MQTs, Sequences, and Application objects. The 'Tables' tab is selected, displaying a table named 'MCL23949.STUDENT'. The table has four columns: USERNAME, EMAIL, Roll number, and PASSWORD. The data rows are as follows:

USERNAME	EMAIL	Roll number	PASSWORD
anu	9	4	6
ayish	3	8	1
ayish	3	8	1

Buttons for 'Back' and 'Export to CSV' are visible in the top right corner of the table view.

2. Performing UPDATE, DELETE Queries with user table:

UPDATE:

The screenshot shows the IBM Db2 on Cloud web interface with the SQL editor open. The script contains the following SQL statements:

```
1 insert into student values('ayish',3,8,1);
2 insert into student values('anu',9,4,6);
3 update student set email=5 where password=1;
```

The 'Run selected' button is highlighted. Below the script editor, the 'History' tab is active, showing the execution history of the script:

Script	Date	Status	Runtime
Untitled - 1	Oct 19, 2022 6:59:52 AM	✓ 1	0.010 s
update student set email=5 where password=1		✓	0.010 s

OUTPUT:

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

bpe61bfd0365e9u4psdglite.db2.cloud.ibm.com/cn%3Av1%3Abluemix%3Apublic%3Adashdb-for-transactions%3Aus-south%3Aa%2F1618829015fa44daa13b06aa2...

IBM Db2 on Cloud

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

MCL23949.STUDENT Back

Export to CSV

USERNAME	EMAIL	Roll number	PASSWORD
anu	9	4	6
ayish	5	8	1
ayish	5	8	1

Type here to search 07:02 19-10-2022

DELETE:

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

bpe61bfd0365e9u4psdglite.db2.cloud.ibm.com/cn%3Av1%3Abluemix%3Apublic%3Adashdb-for-transactions%3Aus-south%3Aa%2F1618829015fa44daa13b06aa2...

IBM Db2 on Cloud

Data objects My script

Filter objects

MCL23949

*Untitled - 1 x + Beta Classic

```
1 insert into student values('ayish',3,8,1);
2 insert into student values('anu',9,4,6);
3 update student set email=5 where password=1;
4 delete from student where email=5;
```

Run selected

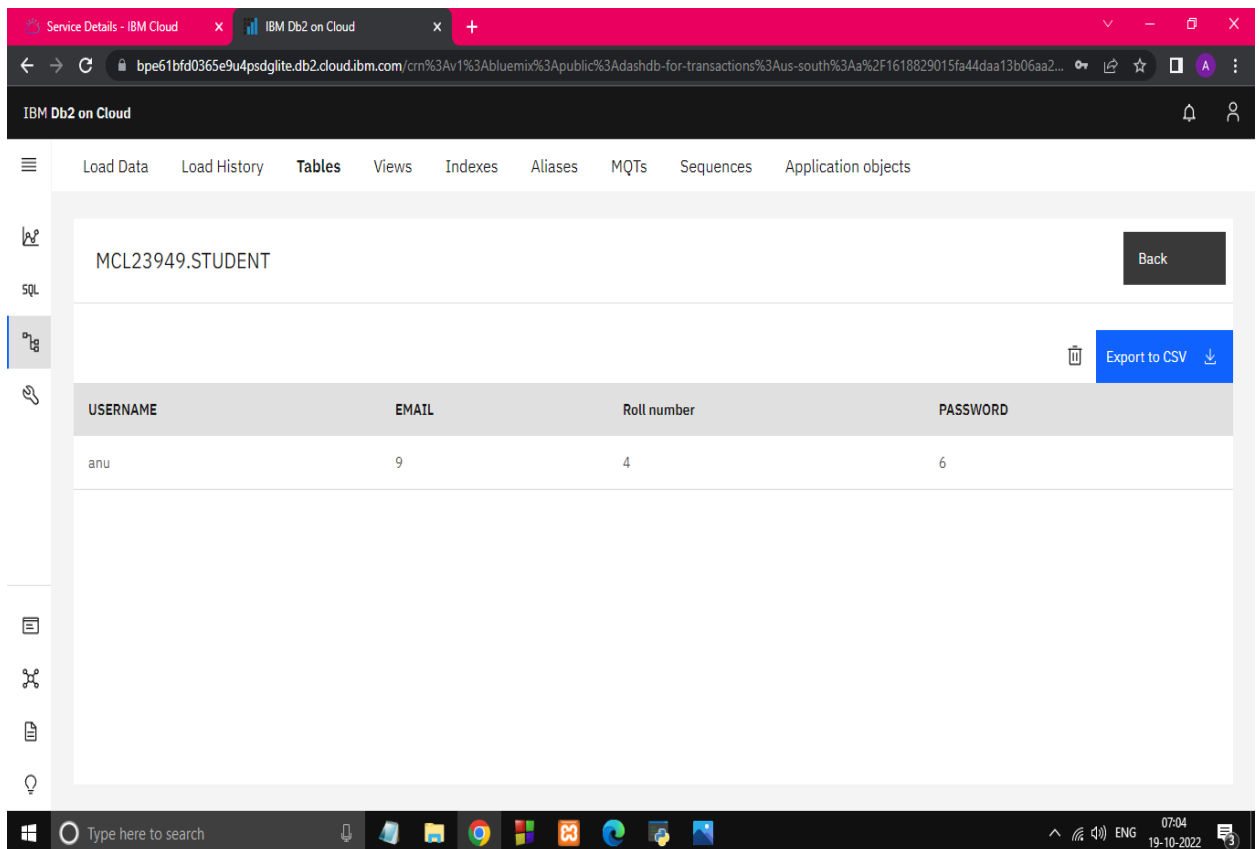
History

Find history

Script	Date	Status	Runtime
Untitled - 1	Oct 19, 2022 7:04:01 AM	1	0.015 s
delete from student where email=5			0.015 s
Untitled - 1	Oct 19, 2022 6:59:52 AM	1	0.010 s
update student set email=5 where password=1			0.010 s

Type here to search 07:04 19-10-2022

OUTPUT:



3. Connect python code to db2:

```
import ibm_db
conn=
ibm_db.connect("DATABASE=bludb;HOSTNAME=9938aec0-8105-433e-8bf9-
0fbb7e483086.clogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=32459;SECURITY=SSL;SSL
ServerCertificate=DigiCertGlobalRoot.crt;UID=mcl23949;PWD=dmUiv2o6zjDMw0Ea",)
print(conn)
print("connection successful...")
```

4. Creating a flask app with registration page, login page and welcome page:

Store this code in 'app.py' file

```
from flask import Flask, render_template, request, redirect, url_for, session
import ibm_db
import re
conn=ibm_db.connect("DATABASE=bludb;HOSTNAME=9938aec0-8105-433e-8bf9-
0fbb7e483086.clogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=32459;SECURITY=SSL;SSL
ServerCertificate=DigiCertGlobalRoot.crt;UID=mcl23949;PWD=dmUiv2o6zjDMw0Ea",)

app = Flask(__name__)
```

```

app.secret_key='a'
conn = ibm_db.connect()
@app.route('/')
@app.route('/login', methods=['GET', 'POST'])
def login():
    msg = ""
    if request.method == 'POST' and 'username' in request.form and 'password' in request.form:
        username = request.form['username']
        password = request.form['password']
        sql=('SELECT * FROM users WHERE username = % s AND password = % s',
        (username, 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['id']
            session['username'] = account['username']
            msg = 'Logged in successfully !'
            return render_template('index.html', msg = msg)
        else:
            msg = 'Incorrect username / password !'
    return render_template('login.html', msg = msg)

@app.route('/logout')
def logout():
    session.pop('loggedin', None)
    session.pop('id', None)
    session.pop('username', None)
    return redirect(url_for('login'))

@app.route('/register', methods=['GET', 'POST'])
def register():
    msg = ""
    if request.method == 'POST' and 'username' in request.form and 'password' in request.form and 'email' in request.form :
        username = request.form['username']
        password = request.form['password']

```

```

        email = request.form['email']
        sql=('SELECT * FROM users WHERE username = % s AND password = % s',
(username, password, ))
        stmt = ibm_db.prepare(conn,sql)
        ibm_db.bind_param(stmt,1,username)
        ibm_db.execute(stmt)
        account = ibm_db.fet_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 = 'Username must contain only characters and numbers !'
        elif not username or not password or not email:
            msg = 'Please fill out the form !'
        else:
            insert_sql=('INSERT INTO accounts VALUES (NULL, % s, % s, % s)',
(username, password, email, ))
            prep_stmt = ibm_db.prepare(conn,insert_sql)
            ibm_db.bind_param(prepare_stmt,1,username)
            ibm_db.bind_param(stmt,2,password)
            ibm_db.bind_param(stmt,3,email)
            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)
if __name__ == '__main__':
    app.run(host='0.0.0.0')

```

register.html:

```

<html>
    <head>
        <meta charset="UTF-8">
        <title> Register </title>
        <link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">
    </head>
    <body style="background-color:powderblue;"></br></br></br></br></br>
        <div align="center">

```

```

<div align="center" class="border">
    <div class="header">
        <h1 class="word">Register</h1>
    </div></br></br></br>
    <h2 class="word">
        <form action="{{ url_for('register') }}" method="post">
            <input id="username" name="username" type="text"
placeholder="Enter Your Username" class="textbox"/></br></br>
            <input id="password" name="password"
type="password" placeholder="Enter Your Password" class="textbox"/></br></br>
            <input id="email" name="email" type="text"
placeholder="Enter Your Email ID" class="textbox"/></br></br>
            <input type="submit" class="btn" value="Sign
Up"></br>
        </form>
    </h2>
    <p class="bottom">Already have an account? <a class="bottom"
href="{{ url_for('login') }}"> Sign In here</a></p>
</div>
</div>
</body>
</html>

```

login.html:

```

<html>
    <head>
        <meta charset="UTF-8">
        <title> Login </title>
        <link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">
    </head>
    <body></br></br></br></br></br>
        <div align="center">
            <div align="center" class="border">
                <div class="header">
                    <h1 class="word">Login</h1>
                </div></br></br></br>
                <h2 class="word">
                    <form action="{{ url_for('login') }}" method="post">
                        <input id="username" name="username" type="text"
placeholder="Enter Your Username" class="textbox"/></br></br>

```



```

                                <input id="password" name="password"
type="password" placeholder="Enter Your Password" class="textbox"/></br></br></br>
                                <input type="submit" class="btn" value="Sign
In"></br></br>

                                </form>
                                </h2>
                                <p class="bottom">Don't have an account? <a class="bottom"
href="{{url_for('register')}}"> Sign Up here</a></p>
                                </div>
                                </div>
                                </body>
                                </html>

```

Index.html:

```

<html>
    <head>
        <meta charset="UTF-8">
        <title> Index </title>
        <link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">

    </head>
    <body></br></br></br></br></br>
        <div align="center">
            <div align="center" class="border">
                <div class="header">
                    <h1 class="word">Index</h1>
                </div></br></br></br>
                <h1 class="bottom">
                    Hi {{session.username}}!!</br></br> Welcome to the
index page...

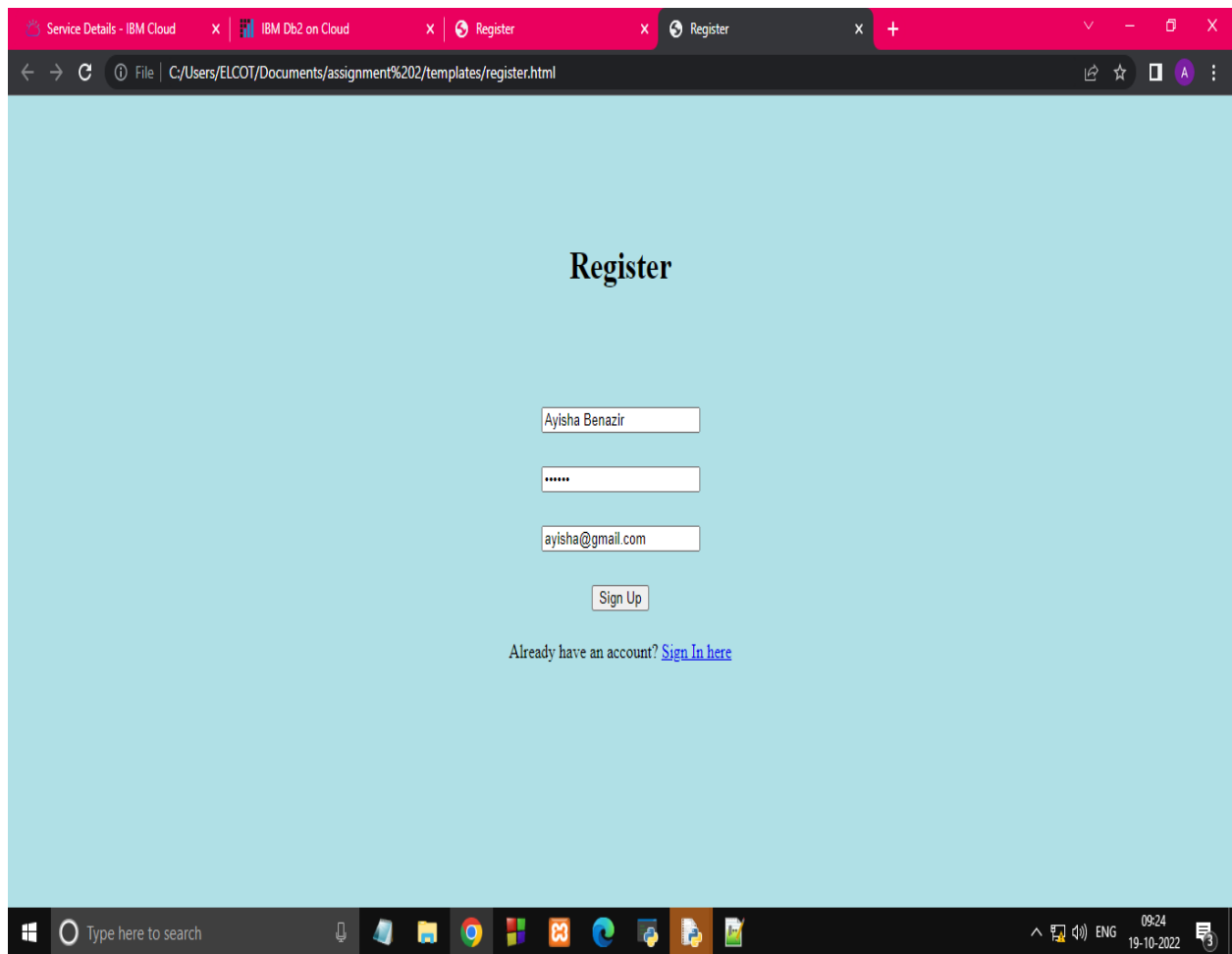
                </h1></br></br></br>
                <a href="{{ url_for('logout') }}" class="btn">Logout</a>

            </div>
        </div>
    </body>
</html>

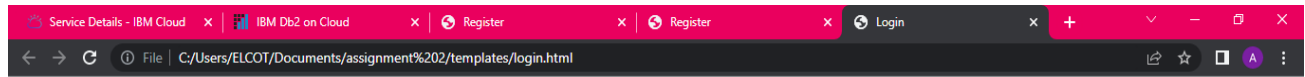
```

OUTPUT:

Register page:



login page:

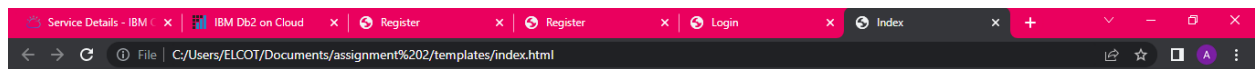


Login

Don't have an account? [Sign Up here](#)



Welcome page:



Index

Hi Ayisha Benazir!!

Welcome to the index page...

[Logout](#)



