

IBM ASSIGNMENT II

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

The screenshot shows the IBM Db2 on Cloud console interface. On the left, the 'Data objects' tab is selected, and the 'My script' section is active. A list of templates is visible, including 'Template - Delete Statement', 'Template - Insert Statement', 'Template - Select Statement', 'Template - SQL Stored Procedure', 'Template - Update Statement', and 'Template - User Defined Function'. The main editor area shows a script titled 'Untitled - 1' with the following SQL code:

```
1 create table User
2 (email varchar(40),
3 username varchar(20),
4 password varchar(20),
5 rollno int
6 );
7
```

Below the script editor, the 'History' tab is selected, showing a table with the following data:

Script	Date	Status	Runtime
Untitled - 1	Sep 25, 2022 7:22:28 PM	1	0.079 s
create table User (email varchar(40), username varchar(20), password varchar(20), rollno int);		1	0.079 s

The screenshot shows the IBM Db2 on Cloud console interface. On the left, the 'Data objects' tab is selected, and the 'My script' section is active. A list of templates is visible, including 'Template - Delete Statement', 'Template - Insert Statement', 'Template - Select Statement', 'Template - SQL Stored Procedure', 'Template - Update Statement', and 'Template - User Defined Function'. The main editor area shows a script titled 'Untitled - 1' with the following SQL code:

```
1 create table User
2 (email varchar(40),
3 username varchar(20),
4 password varchar(20),
5 rollno int
6 );
7
8 insert into User(email,username,password,rollno) values ('nn@gmail.com','Nikki','nikki',40);
```

Below the script editor, the 'History' tab is selected, showing a table with the following data:

Script	Date	Status	Runtime
Untitled - 1	Sep 25, 2022 7:32:43 PM	1	0.006 s
insert into User(email,username,password,rollno) values ('nn@gmail.com','Nikki','nikki',40);		1	0.006 s
Untitled - 1	Sep 25, 2022 7:32:02 PM	1	0.008 s
insert into User(email,username,password,rollno) values ('nn','Nikhila','nikhila',40);		1	0.008 s
Untitled - 1	Sep 25, 2022 7:30:54 PM	1	0.013 s
Untitled - 1	Sep 25, 2022 7:28:40 PM	1	0.025 s
Untitled - 1	Sep 25, 2022 7:26:12 PM	1	0.015 s
Untitled - 1	Sep 25, 2022 7:22:28 PM	1	0.079 s

2. Perform UPDATE,DELETE Queries with user table

The screenshot shows the IBM Db2 on Cloud interface. A SQL script is executed, and the results are displayed in a table. The script includes a CREATE TABLE statement, an INSERT statement, and a SELECT statement. The results table shows two rows of data.

```
1 create table User
2 (email varchar(40),
3  username varchar(20),
4  password varchar(20),
5  rollno int
6 );
7
8 insert into User(email,username,password,rollno) values ('nn@gmail.com','Nikhi','nikki',40);
9
10 select * from User;
```

EMAIL	USERNAME	PASSWORD	ROLLNO
nn	Nikhila	nikhi	4040
nn@gmail.com	Nikhi	nikki	40

The screenshot shows the IBM Db2 on Cloud interface. A SQL script is executed, and the results are displayed in a table. The script includes a CREATE TABLE statement, an INSERT statement, an UPDATE statement, and a SELECT statement. The results table shows two rows of data, with the email address updated for the user with rollno 4040.

```
1 (email varchar(40),
2  username varchar(20),
3  password varchar(20),
4  rollno int
5 );
6
7 insert into User(email,username,password,rollno) values ('nn@gmail.com','Nikhi','nikki',40);
8
9 select * from User;
10
11 update User set email='nikhi@gmail.com' where rollno=4040;
12
13 select * from User;
```

EMAIL	USERNAME	PASSWORD	ROLLNO
nikhi@gmail.com	Nikhila	nikhi	4040
nn@gmail.com	Nikhi	nikki	40

IBM Db2 on Cloud

No statement is selected. Please select one or more statements and try again, or select Run All.

Filter scripts

Template - Delete Statement
Template - Insert Statement
Owner System
Category Templates
Size 0B
Saved 7/31/2020
Script INSERT INTO DB2ADMIN.T...
[Show more](#)

Template - Select Statement
Template - SQL Stored Procedure
Template - Update Statement
Template - User Defined Function

```
4 password varchar(20),
5 rollno int
6 );
7
8 insert into User(email,username,password,rollno) values ('nn@gmail.com','Nikki','nikki',40);
9 insert into User(email,username,password,rollno) values ('nk@gmail.com','Nikhileshwari','ibm',18);
10
11 select * from User;
12
```

History Results

Result set 1 Details

Filter table Total:3

EMAIL	USERNAME	PASSWORD	ROLLNO
nikhi@gmail.com	Nikhila	nikhi	4040
nn@gmail.com	Nikki	nikki	40
nk@gmail.com	Nikhileshwari	ibm	18

IBM Db2 on Cloud

No statement is selected. Please select one or more statements and try again, or select Run All.

Filter scripts

Template - Delete Statement
Template - Insert Statement
Owner System
Category Templates
Size 0B
Saved 7/31/2020
Script INSERT INTO DB2ADMIN.T...
[Show more](#)

Template - Select Statement
Template - SQL Stored Procedure
Template - Update Statement
Template - User Defined Function

```
8 insert into User(email,username,password,rollno) values ('nn@gmail.com','Nikki','nikki',40);
9 insert into User(email,username,password,rollno) values ('nk@gmail.com','Nikhileshwari','ibm',18);
10
11 select * from User;
12
13 update User set email='nikhi@gmail.com' where rollno=4040;
14
15 select * from User;
16
17 delete from User where rollno=4040;
```

History Results

Run time 0.006 s
Run by tbl66129
Database cmv1:bluemix:public:dashdb-for-transactions:us-south-a/9b11ea794bef4fa39b0494c5ca83faef3ad19f5300-a7a8-4481-9b23-54a60748f22d:
Affected rows 1

Full query body

```
delete from User where rollno=4040;
```

IBM Db2 on Cloud

No statement is selected. Please select one or more statements and try again, or select Run All.

Filter scripts

Template - Delete Statement
Template - Insert Statement
Owner System
Category Templates
Size 0B
Saved 7/31/2020
Script INSERT INTO DB2ADMIN.T...
[Show more](#)

Template - Select Statement
Template - SQL Stored Procedure
Template - Update Statement
Template - User Defined Function

```
8 insert into User(email,username,password,rollno) values ('nn@gmail.com','Nikki','nikki',40);
9 insert into User(email,username,password,rollno) values ('nk@gmail.com','Nikhileshwari','ibm',18);
10
11 select * from User;
12
13 update User set email='nikhi@gmail.com' where rollno=4040;
14
15 select * from User;
16
17 delete from User where rollno=4040;
```

History Results

Result set 1 Details

Filter table Total:2

EMAIL	USERNAME	PASSWORD	ROLLNO
nn@gmail.com	Nikki	nikki	40
nk@gmail.com	Nikhileshwari	ibm	18

3. Connect python code to db2

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=bludb;HOSTNAME=6667d8e9-9d4d-4ccb-ba32-
21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=30376;SECURITY
=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=tbl66129;PWD=LPzyHjRGcm
XprQoN", "", "")

@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)

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

```

register.html

```

<!DOCTYPE html>

<html>

<body>

<center><h1>ASSIGNMENT 2</h1></center>

<center><h2>REGISTRATION</h2></center>

<form action="http://localhost:5000" method="POST">

<p>Enter username:</p>

<p><input type="text" name="username"/></p>

<p>Enter mail:</p>

<p><input type="email" name="email"/></p>

<p>Enter password:</p>

<p><input type="password" name="password"/></p>

<p><input type="submit" value="Submit"/></p>

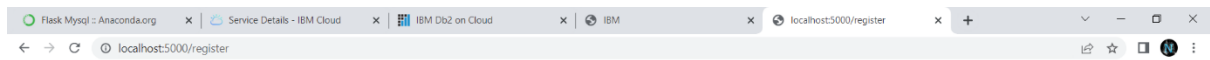
{{ msg }}

</form>

</body>

</html>

```



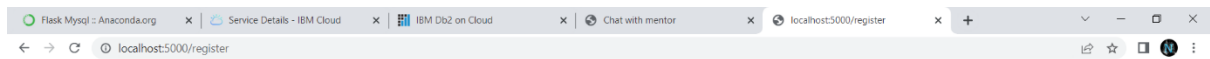
ASSIGNMENT 2

Connecting python to db2

Enter username:

Enter mail:

Enter password:



ASSIGNMENT 2

Connecting python to db2

Enter username:

Enter mail:

Enter password:

You have successfully registered !



IBM Db2 on Cloud

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

TBL66129.USERS

Back

Export to CSV

USERNAME	EMAIL	PASSWORD
Harry	harry@gmail.com	harry
Nikhi	nik@abc.com	nikabc
Nikhila	nikhi@mail.com	nikhila
abc	abc@abc.com	abc

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

Register.html

```
<!DOCTYPE html>

<html>

<body>

<center><h1>ASSIGNMENT 2</h1></center>

<center><h2>REGISTRATION</h2></center>

<form action="http://localhost:5000" method="POST">

<p>Enter username:</p>

<p><input type="text" name="username"/></p>

<p>Enter mail:</p>

<p><input type="email" name="email"/></p>

<p>Enter password:</p>

<p><input type="password" name="password"/></p>

<p><input type="submit" value="Submit"/></p>

{{ msg }}

</form>
```

```
</body>
```

```
</html>
```

Login.html

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<center><h1>ASSIGNMENT 2</h1></center>
```

```
<center><h2>LOGIN</h2></center>
```

```
<form action="http://localhost:5000/login" method="POST">
```

```
<p>Enter username:</p>
```

```
<p><input type="text" name="username"/></p>
```

```
<p>Enter password:</p>
```

```
<p><input type="password" name="password"/></p>
```

```
<p><input type="submit" value="Submit"/></p>
```

```
{{msg}}
```

```
</form>
```

```
</body>
```

```
</html>
```

Welcome.html

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h1>WELCOME</h1>
```

```
<form action="http://localhost:5000/welcome" methods="get">
```

```
<h4>You have successfully logged in!</h4>
```

```
</form>
```

```
{{msg}}
```

```
</body>
```

```
</html>
```

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=bludb;HOSTNAME=6667d8e9-9d4d-4ccb-ba32-
21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=30376;SECURITY
=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=tbl66129;PWD=LPzyHjRGcm
XprQoN",",")

@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:

            print("hi")

            session['loggedin'] = True

            session['id'] = account['USERNAME']

            userid= account['USERNAME']

            session['username'] = account['USERNAME']

            msg = 'Logged in successfully !'

            return render_template('login.html', msg = msg)

```

```

else:

    msg = 'Incorrect username / password !'

return render_template('login.html', msg = msg)
@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 !'

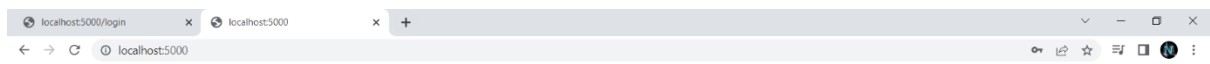
    elif request.method == 'POST':

        msg = 'Please fill out the form !'

    return render_template('register.html', msg = msg)

if __name__ == '__main__':

    app.run(debug = True)
```



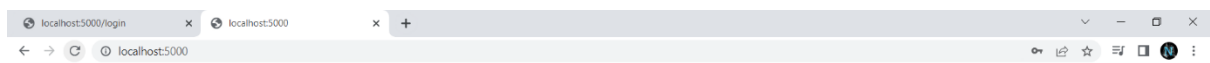
ASSIGNMENT 2

REGISTRATION

Enter username:

Enter mail:

Enter password:



ASSIGNMENT 2

REGISTRATION

Enter username:

Enter mail:

Enter password:

You have successfully registered !

Service Details - IBM Cloud x IBM Db2 on Cloud x Redirect Parameters Flask With C x Solved: My WiFi options disapp x Fix Network Adapter Not Showin x +

bpe61bfd0365e9u4psdglite.db2.cloud.ibm.com/crn%3Av1%3Abluemix%3Apublic%3Adashdb-for-transactions%3Aus-south%3Aa%2F9b11ea794bef4fa39b0494c5ca83ae%3Ad19f5300-a7a8-...

IBM Db2 on Cloud

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

TBL66129.USERS Back

Export to CSV

USERNAME	EMAIL	PASSWORD
Donna	donna@abc.com	donna
Harry	harry@gmail.com	harry
Nikhi	nik@abc.com	nikabc
Nikhila	nikhi@mail.com	nikhila
Nikhileshwari	nikhis@gmail.com	nikhila234
abc	abc@abc.com	abc

localhost:5000/register x localhost:5000/login x +

localhost:5000/login

ASSIGNMENT 2

LOGIN

Enter username:

Enter password:

Submit

localhost:5000/welcome x TypeError: welcome() missing 1 x +

localhost:5000/welcome

WELCOME

You have successfully logged in!

IBM Db2 on Cloud

Load DataLoad HistoryTablesViewsIndexesAliasesMQTsSequencesApplication objects

TBL66129.USERS

Back

Export to CSV

USERNAME	EMAIL	PASSWORD
Donna	donna@abc.com	donna
Harry	harry@gmail.com	harry
Nikhi	nik@abc.com	nikabc
Nikhila	nikhi@mail.com	nikhila
Nikhileshwari	nikhis@gmail.com	nikhila234
abc	abc@abc.com	abc

localhost:5000/login

localhost:5000

localhost:5000/login

ASSIGNMENT 2

LOGIN

Enter username:

Enter password:

Incorrect username / password !