

ASSIGNMENT – 2

Student Name	Jaimugil C
Student Roll number	311519104023
Team ID	PNT2022TMID27815

1. CREATE USER TABLE WITH USER WITH EMAIL, USERNAME, ROLL NUMBER, PASSWORD

TABLE CREATION:

The screenshot shows the IBM Db2 on Cloud web interface. The 'Tables' tab is selected, and the 'REGISTRATION' table is highlighted. The table definition is displayed on the right side of the interface.

Name	Data type	Nullable	Length	Scale
EMAIL	VARCHAR	Y	50	0
USERNAME	VARCHAR	Y	50	0
ROLL_NO	INTEGER	Y		0
PASSWORD	VARCHAR	Y	50	0

The screenshot shows the IBM Db2 on Cloud web interface with the SQL editor and the execution results. The SQL script is executed, and the results are displayed in the 'History' tab.

```
1 insert into registration values('sha@gmail.com','sha',2,'shap');
2 insert into registration values('mug@gmail.com','mug',3,'mug');
3 insert into registration values('nar@gmail.com','nar',4,'nar');
```

Script	Date	Status	Runtime
Untitled - 1	Oct 30, 2022 4:22:27 PM	3	0.018 s
insert into registration values('sha@gmail.com','sha',2,'shap')		✓	0.006 s
insert into registration values('mug@gmail.com','mug',3,'mug')		✓	0.008 s
insert into registration values('nar@gmail.com','nar',4,'nar')		✓	0.004 s
Untitled - 1	Oct 30, 2022 4:20:26 PM	1	0.007 s
insert into registration values('dan@gmail.com','dan',1,'danp')		✓	0.007 s
Untitled - 1	Oct 30, 2022 4:19:58 PM	1	0.006 s
insert into users values('dan@gmail.com','dan',1,'danp')		✗	0.006 s

IBM Db2 on Cloud

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

NYC06722.REGION

Back

Export to CSV

EMAIL	USERNAME	ROLL_NO	PASSWORD
dan@gmail.com	dan	1	danp
mug@gmail.com	mug	3	mug
nar@gmail.com	nar	4	nar
sha@gmail.com	sha	2	shap

2. PERFORM UPDATE, DELETE QUERIES WITH USER TABLE

UPDATE QUERY:

IBM Db2 on Cloud

Data objects Saved objects

Filter objects

NYC06722

Tables

REGISTRATION

Views

MQTs

Aliases

Nicknames

*Untitled - 1

```

1 insert into registration values('sha@gmail.com','sha',2,'shap');
2 insert into registration values('mug@gmail.com','mug',3,'mug');
3 insert into registration values('nar@gmail.com','nar',4,'nar');
4
5 update registration set ROLL_NO=20 where username='nar';

```

Syntax assistant

Run selected

History

Find history

Script	Date	Status	Runtime
Untitled - 1	Oct 30, 2022 4:25:21 PM	✓ 1	0.005 s
update registration set ROLL_NO=20 where username='nar'		✓	0.005 s
Untitled - 1	Oct 30, 2022 4:24:24 PM	✗ 1	0.007 s
update registration set roll_no=20 where usern		✗	0.007 s
Untitled - 1	Oct 30, 2022 4:23:51 PM	✗ 1	0.006 s
update registration set rollno=20 where username=nar		✗	0.006 s
Untitled - 1	Oct 30, 2022 4:23:46 PM	✗ 1	0.008 s
update registration set rollno=20 where username=nar		✗	0.008 s
Untitled - 1	Oct 30, 2022 4:22:27 PM	✓ 3	0.018 s
insert into registration values('sha@gmail.com','sha',2,'shap')		✓	0.006 s

IBM Db2 on Cloud

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

NYC06722.REGISTRATION

Export to CSV

EMAIL	USERNAME	ROLL_NO	PASSWORD
dan@gmail.com	dan	1	danp
mug@gmail.com	mug	3	mug
nar@gmail.com	nar	20	nar
sha@gmail.com	sha	2	shap

DELETE QUERY:

IBM Db2 on Cloud

Data objects Saved objects

Filter objects

NYC06722

Tables

REGISTRATION

Views

MQTs

Aliases

Nicknames

*Untitled - 1

```
2 insert into registration values('mug@gmail.com','mug',3,'mug');
3 insert into registration values('nar@gmail.com','nar',4,'nar');
4
5 update registration set ROLL_NO=20 where username='nar';
6
7 delete from registration where roll_no=3;
```

History

Find history

Script	Date	Status	Runtime
Untitled - 1	Oct 30, 2022 4:26:33 PM	✓ 1	0.005 s
delete from registration where roll_no=3		✓	0.005 s
Untitled - 1	Oct 30, 2022 4:25:21 PM	✓ 1	0.005 s
update registration set ROLL_NO=20 where username='nar'		✓	0.005 s
Untitled - 1	Oct 30, 2022 4:24:24 PM	✗ 1	0.007 s
update registration set roll_no=20 where userna		✗	0.007 s
Untitled - 1	Oct 30, 2022 4:23:51 PM	✗ 1	0.006 s
update registration set rollno=20 where username=nar		✗	0.006 s
Untitled - 1	Oct 30, 2022 4:23:46 PM	✗ 1	0.008 s
update registration set rollno=20 where username=nar		✗	0.008 s

IBM Db2 on Cloud

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

NYC06722.REGISTRATION Back

Export to CSV

EMAIL	USERNAME	ROLL_NO	PASSWORD
dan@gmail.com	dan	1	danp
nar@gmail.com	nar	20	nar
sha@gmail.com	sha	2	shap

3. CONNECT PYTHON TO DB2

CONNECTING TO DB:

ass.py

```
from flask import Flask, render_template, redirect, url_for, request, session
import ibm_db
import re
```

```
app = Flask(__name__)
app.secret_key = 'a'
```

```
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=32733;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt; UID=rrj92864; PWD=maDr7SBpdojnrgv;", "", "")
```

```
@app.route('/')
def home():
```

```
    return render_template('reg2.html')
```

```
@app.route('/login', methods=["GET", "POST"])
```

```
def login():
```

```
    global userid
    msg = ""
```

```
    if request.method == "POST":
```

```
        username = request.form['uname']
```

```
        password = request.form['pwd']
```

```
        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['USERNAME']
    userid = account["USERNAME"]
    session['username'] = account["USERNAME"]
    msg = 'Logged in successfully!'
    return redirect(url_for('welcome', username=username))
else:
    msg = "Incorrect Username/Password"
    return render_template('login.html', msg=msg)

```

```
@app.route('/reg2', methods=["GET", "POST"])
```

```
def registration():
```

```
    msg = " "
```

```
    if request.method == "POST":
```

```
        username = request.form['uname']
```

```
        email = request.form['email']
```

```
        password = request.form['pwd']
```

```
        rollno = request.form['rollno']
```

```
        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 = "Username must contain only alphabets and numbers."
```

```
        else:
```

```
            insert_sql = "INSERT INTO USERS VALUES(?,?,?,?)"
```

```
            prep_stmt = ibm_db.prepare(conn, insert_sql)
```

```
            ibm_db.bind_param(prepare_stmt, 1, email)
```

```
            ibm_db.bind_param(prepare_stmt, 2, username)
```

```
            ibm_db.bind_param(prepare_stmt, 3, rollno)
```

```
            ibm_db.bind_param(prepare_stmt, 4, password)
```

```
            ibm_db.execute(prepare_stmt)
```

```
            msg = "You have successfully registered."
```

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

```
    elif request.method == 'POST':
```

```
        msg = "Please fill out the form."
```

```
        return render_template('reg2.html', msg=msg)
```

```
@app.route('/welcome/<username>')
def welcome(username):
    return "Welcome %s!" % username

if __name__ == "__main__":
    app.run(host='0.0.0.0')
```

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

reg.html (Registration Page):-

```
<html>
<body>
    <center>
        <h3><b>REGISTRATION FOR PLASMA DONATION</b></h3>
        <form action = "http://localhost:5000/reg2" method = "POST">
            <p>Enter Register No.: <input type = "text" name="rollno"/></p>
            <p>Enter Email ID: <input type = "text" name="email"/></p>
            <p>Enter Username: <input type = "text" name="uname"/></p>
            <p>Enter Password: <input type = "password" name="pwd"/></p>
            <p><input type = "submit" value="SUBMIT"/></p>
        </form>
    </center>
</body>
</html>
```

login.html (Login Page):-

```
<html>
<body>
    <center>
        <h3><b>LOGIN</b></h3>
        <form action = "http://localhost:5000/login" method = "POST">
            <p>Enter Username: <input type = "text" name="uname"/></p>
            <p>Enter Password: <input type = "password" name="pwd"/></p>
            <p><input type = "submit" value="SUBMIT"/></p>
        </form>
    </center>
</body>
</html>
```

Registration page:

127.0.0.1:5000

REGISTER FOR PLASMA DONATION

Enter Register No.:

Enter Email ID:

Enter Username:

Enter Password:

Activate Windows
Go to PC settings to activate Windows.

Login Page:

localhost:5000/reg2

LOGIN

Enter Username:

Enter Password:

Activate Windows
Go to PC settings to activate Windows.

Welcome Page:

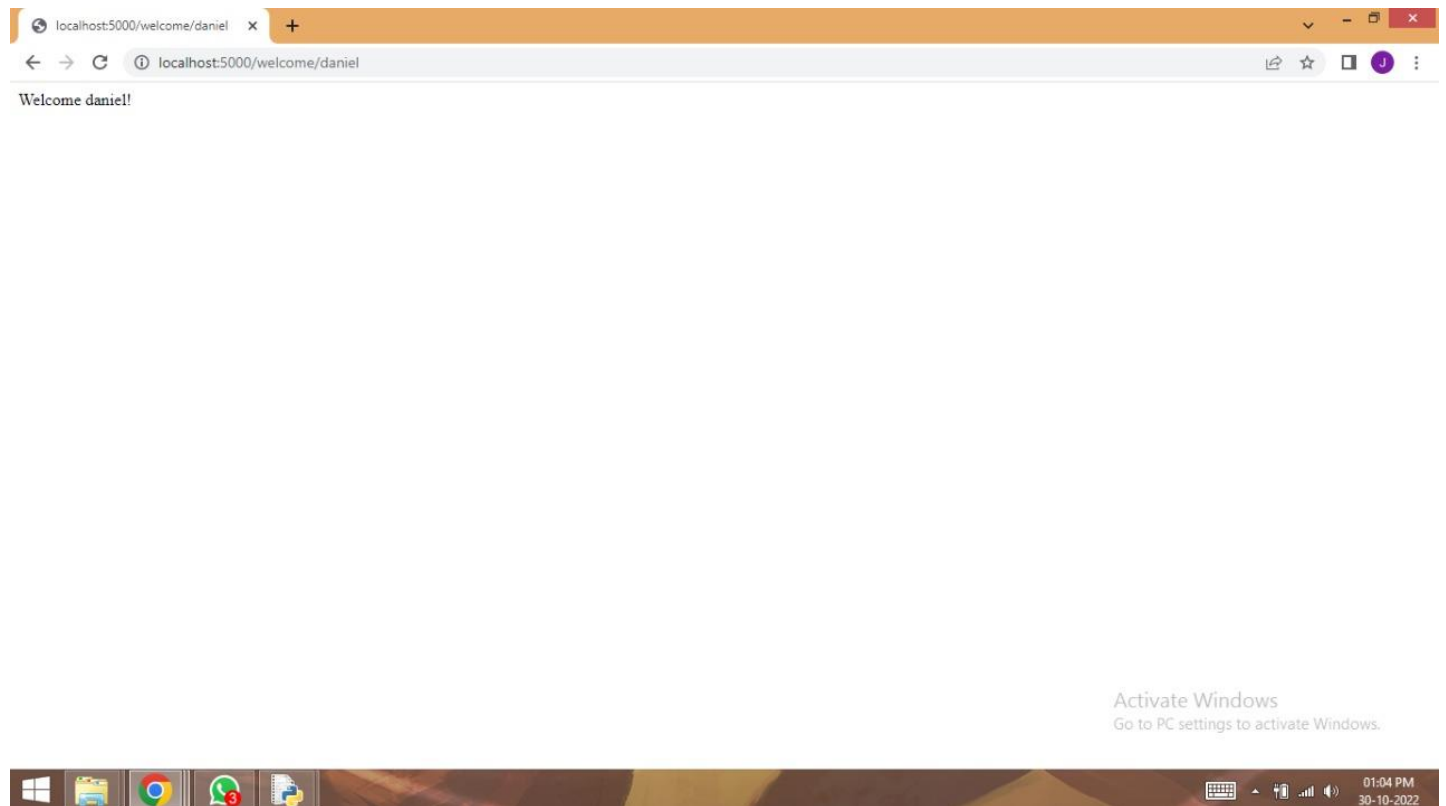


Table Insertion:

