

## ASSIGNMENT – II

**1, 2: - Create user table with email USERNAME ,ROLL\_NO, PASSWORD and perform insert ,update and delete.**

### QUERY:

```
create table user(username varchar(255),email varchar(255), roll_no
varchar(255),password varchar(255));
insert into user
values('arun','arun@01','001','arun@01'),('jeeva','jeeva@gmail.com','002','jeeva@02'),
('aruna','aruna@gmail.com','008','aruna@08'),('dinesh','diasfh@gmail.com','024','Poio
y@24');
select * from user;
update user set email='angel@gmail.com' where roll_no='008';
insert into user values('dineshs','dineshs@gmail.com','025','Omfs@025');
delete user where roll_no='024';
select * from user;
```

The screenshot displays the IBM Db2 on Cloud web interface. The top section shows a browser window with the URL `bpe61bfd0365e9u4psdglite.db2.cloud.ibm.com`. The interface includes a sidebar with navigation options like 'Data objects', 'My script', and 'SQL'. The main area shows a SQL script in a text editor with the following code:

```

1 create table user(username varchar(255),email varchar(255), roll_no varchar(255),password varchar(255));
2 insert into user values('arun','arun@01','001','jeeva','jeeva@gmail.com','002','jeeva@02'),('aruna','aruna@08');
3 select * from user;
4 update user set email='angel@gmail.com' where roll_no='008';
5 insert into user values('dineshs','dineshs@gmail.com','025','Omfs@025');
6 delete user where roll_no='024';
7 select * from user;
8

```

Below the script, there is a 'History' tab. The bottom section of the interface shows the 'Tables' view for the database 'PQW81844.USER'. It displays a table with the following data:

USERNAME	EMAIL	ROLL_NO	PASSWORD
arun	arun@01	001	arun@01
arun	arun@01	001	arun@01
aruna	angel@gmail.com	008	aruna@08
aruna	angel@gmail.com	008	aruna@08
dineshs	dineshs@gmail.com	025	Omfs@025
jeeva	jeeva@gmail.com	002	jeeva@02
jeepva	jeepva@gmail.com	002	jeepva@02

### 3. Connect python to db2

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

```

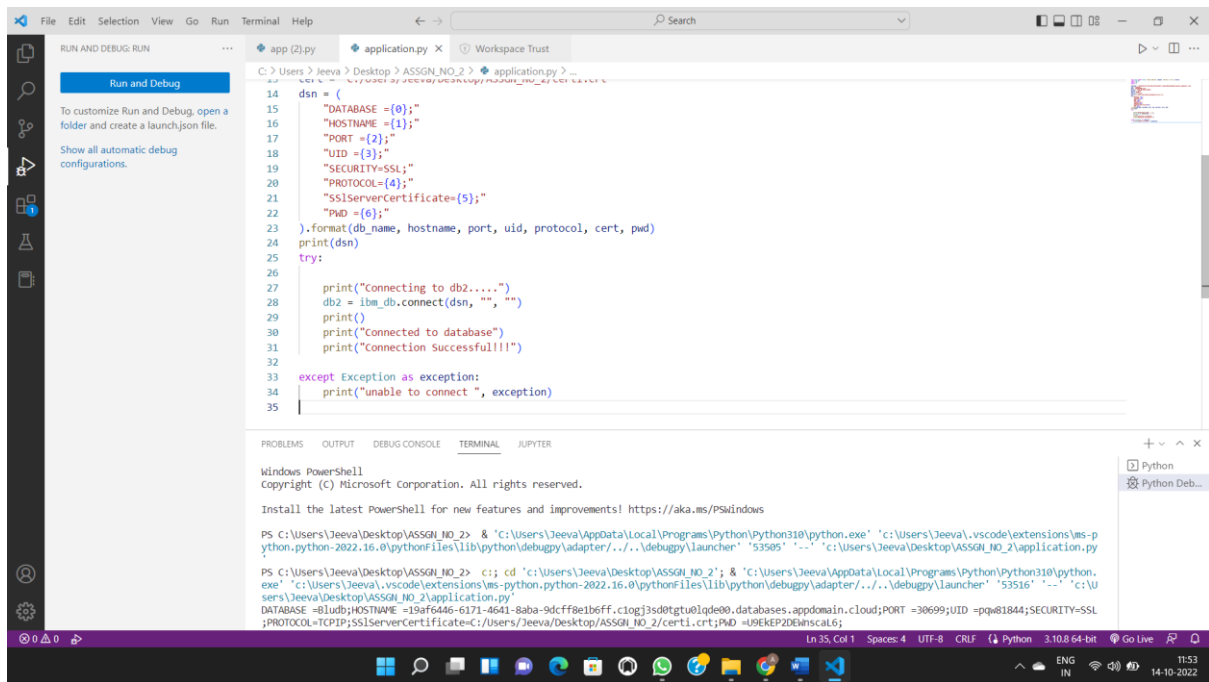
import ibm_db
import db2
import re

hostname = '1bbf73c5-d84a-4bb0-85b9-
ab1a4348f4a4.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud'
uid = 'mwr93891'
pwd = 'gcXyQeZ9NFFyxFwM'
driver = "{IBM DB2 ODBC DRIVER}"
db_name = 'Bludb'
port = '30699'
protocol = 'TCPIP'
cert = "C:/Users/Jeeva/Desktop/ASSGN_NO_2/certi.crt"
dsn = (
    "DATABASE ={0};"
    "HOSTNAME ={1};"
    "PORT ={2};"
    "UID ={3};"
    "SECURITY=SSL;"
    "PROTOCOL={4};"
    "SSLServerCertificate={5};"
    "PWD ={6};"
).format(db_name, hostname, port, uid, protocol, cert, pwd)
print(dsn)
try:

    print("Connecting to db2.....")
    db2 = ibm_db.connect(dsn, "", "")
    print()
    print("Connected to database")
    print("Connection Successful!!!")

except Exception as exception:
    print("unable to connect ", exception)

```



## 4) ACCESS LOGIN WITH CONNTING TO DATABASE

```

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

```

```

app = Flask(__name__)

```

```

hostname = '1bbf73c5-d84a-4bb0-85b9-
ab1a4348f4a4.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud'
uid = 'mwr93891'
pwd = 'gcXyQeZ9NFFyxFWM'
driver = "{IBM DB2 ODBC DRIVER}"
db_name = 'Bludb'
port = '30699'
protocol = 'TCPIP'
cert = "C:/Users/Jeeva/Desktop/ASSGN_NO_2/certi.crt"
dsn = (
    "DATABASE={0};"
    "HOSTNAME={1};"
    "PORT={2};"
    "UID={3};"
    "SECURITY=SSL;"
    "PROTOCOL={4};"
    "SSLServerCertificate={5};"
    "PWD={6};"
).format(db_name, hostname, port, uid, protocol, cert, pwd)
connection = ibm_db.connect(dsn, "", "")

```

```

print()
# query = "SELECT username FROM USER1 WHERE username=?"
# stmt = ibm_db.prepare(connection, query)
# ibm_db.bind_param(stmt, 1, username)
# ibm_db.execute(stmt)
# username = ibm_db.fetch_assoc(stmt)
# print(username)
app.secret_key = 'a'

@app.route('/', methods=['GET', 'POST'])
@app.route('/register', methods=['GET', 'POST'])
def register():
    msg = " "
    if request.method == 'POST':
        username = request.form['username']
        email_id = request.form['email_id']
        phone_no = request.form['phone_no']
        password = request.form['password']
        query = "SELECT * FROM USER1 WHERE username=?;"
        stmt = ibm_db.prepare(connection, query)
        ibm_db.bind_param(stmt, 1, username)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
        if (account):

            msg = "Account already exists!"
            return render_template('register.html', msg=msg)
        # elif not re.match(r'^@+@[^@]+\.[^@]+', email_id):
        #     msg = "Invalid email address"
        # elif not re.match(r'[A-Za-z0-9+]', username):
        #     msg = "Name must contain only characters and numbers"
        else:
            query = "INSERT INTO USER1 values(?,?,?,?)"
            stmt = ibm_db.prepare(connection, query)
            ibm_db.bind_param(stmt, 1, username)
            ibm_db.bind_param(stmt, 2, email_id)
            ibm_db.bind_param(stmt, 3, phone_no)
            ibm_db.bind_param(stmt, 4, password)
            ibm_db.execute(stmt)
            msg = 'You have successfully Logged In!!'
            return render_template('login.html', msg=msg)
    else:
        msg = 'PLEASE FILL OUT OF 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']
    query = "select * from user1 where username=? and password=?"
    stmt = ibm_db.prepare(connection, query)
    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']
        session['username'] = account['USERNAME']
        msg = 'Logged in Successfully'
        return render_template('welcome.html', msg=msg,
username=str.upper(username))
    else:
        msg = 'Incorrect Username or Password'
        return render_template('login.html', msg=msg)
else:
    msg = 'PLEASE FILL OUT OF THE FORM'
    return render_template('login.html', msg=msg)

@app.route('/welcome', methods=['GET', 'POST'])
def welcome():
    if request.method == 'POST':
        username = request.form['username']
        print(username)
        return render_template('welcome.html', username=username)
    else:
        return render_template('welcome.html', username=username)

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

```