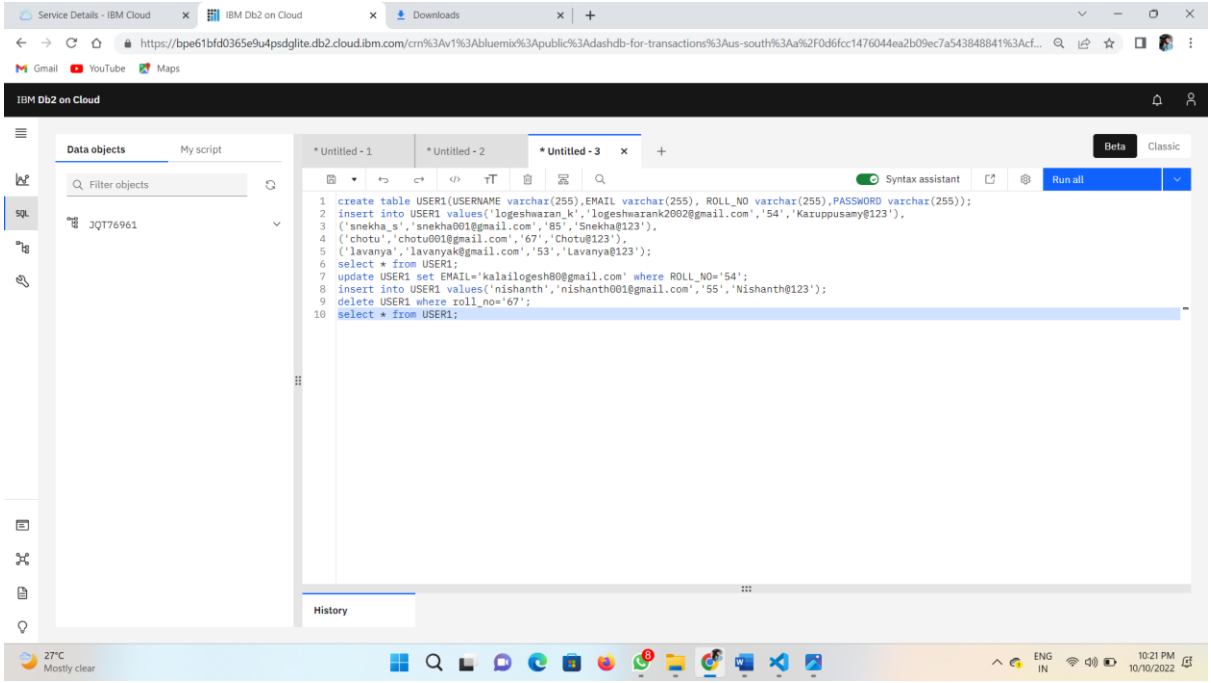
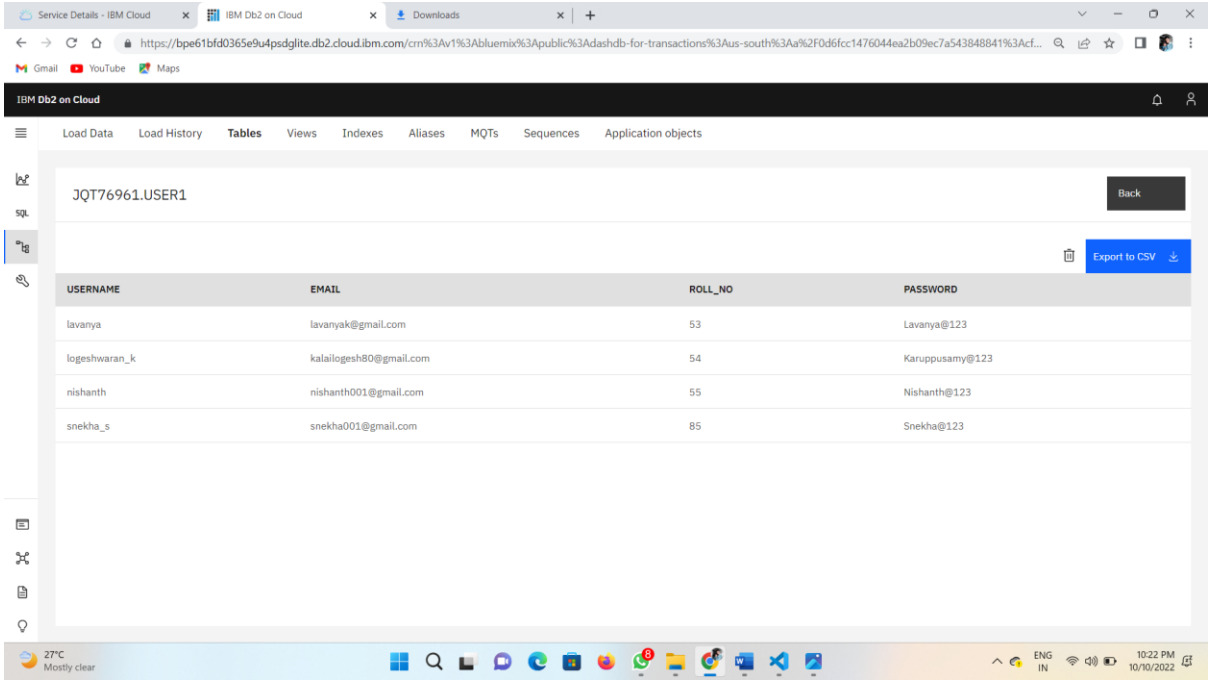
ASSIGNMENT – II

1, 2: - Create user table with email USERNAME ,ROLL_NO, PASSWORD and perform insert ,update and delete.

QUERY:

```
create table USER1(USERNAME varchar(255),EMAIL varchar(255), ROLL_NO
varchar(255),PASSWORD varchar(255));
insert into USER1
values('logeshwaran_k','logeshwarank2002@gmail.com','54','Karuppusamy@123'),
('snekha_s','snekha001@gmail.com','85','Snekha@123'),
('chotu','chotu001@gmail.com','67','Chotu@123'),
('lavanya','lavanyak@gmail.com','53','Lavanya@123');
select * from USER1;
update USER1 set EMAIL='kalailogesh80@gmail.com' where ROLL_NO='54';
insert into USER1 values('nishanth','nishanth001@gmail.com','55','Nishanth@123');
delete USER1 where roll_no='67';
select * from USER1;
```





3. Connect python to db2

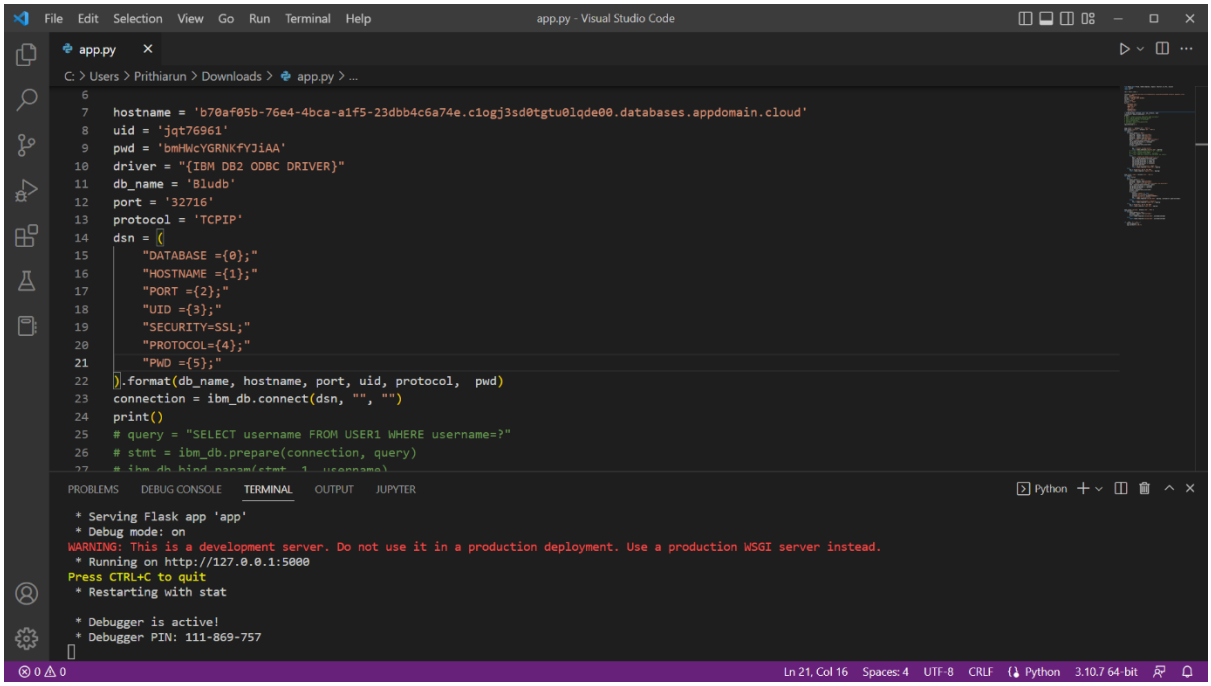
```
from flask import Flask, render_template, request, redirect, url_for, session
import ibm_db
import re

hostname = 'b70af05b-76e4-4bca-a1f5-
23dbb4c6a74e.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud'
uid = 'jqt76961'
pwd = 'bmHWcYGRNKfYJiAA'
driver = "{IBM DB2 ODBC DRIVER}"
db_name = 'Bludb'
port = '32716'
protocol = 'TCPIP'
cert="C:/Users/loges/OneDrive/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
import re

app = Flask(__name__)

hostname = 'b70af05b-76e4-4bca-a1f5-
23dbb4c6a74e.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud'
uid = 'jqt76961'
pwd = 'bmHWcYGRNKfYJiAA'
driver = "{IBM DB2 ODBC DRIVER}"
db_name = 'Bludb'
port = '32716'
protocol = 'TCPIP'
cert="C:/Users/loges/OneDrive/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 addres"
        # 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')
```