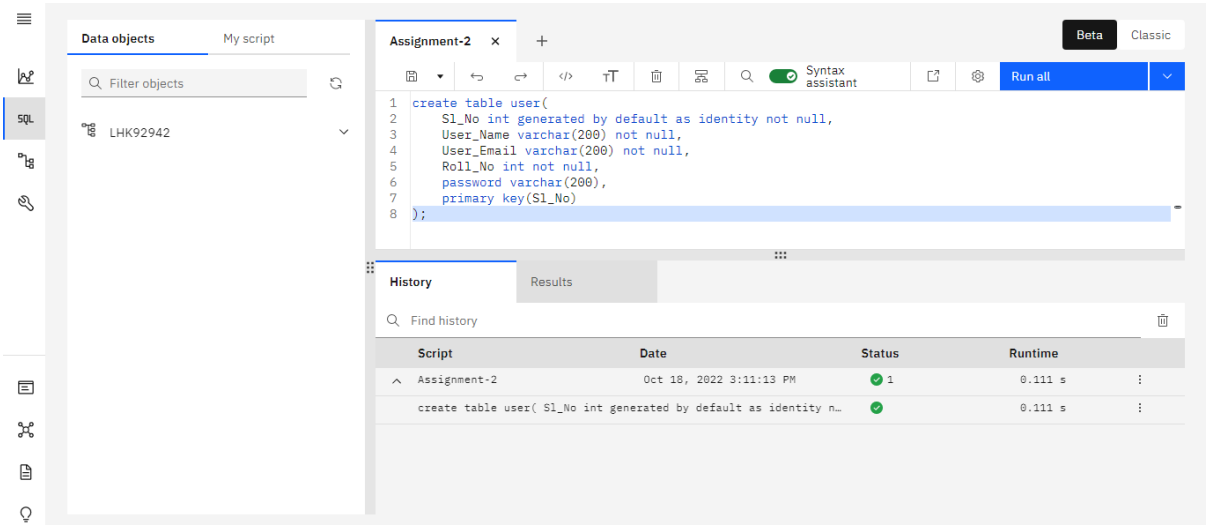
Assignment -2

DB and Flask

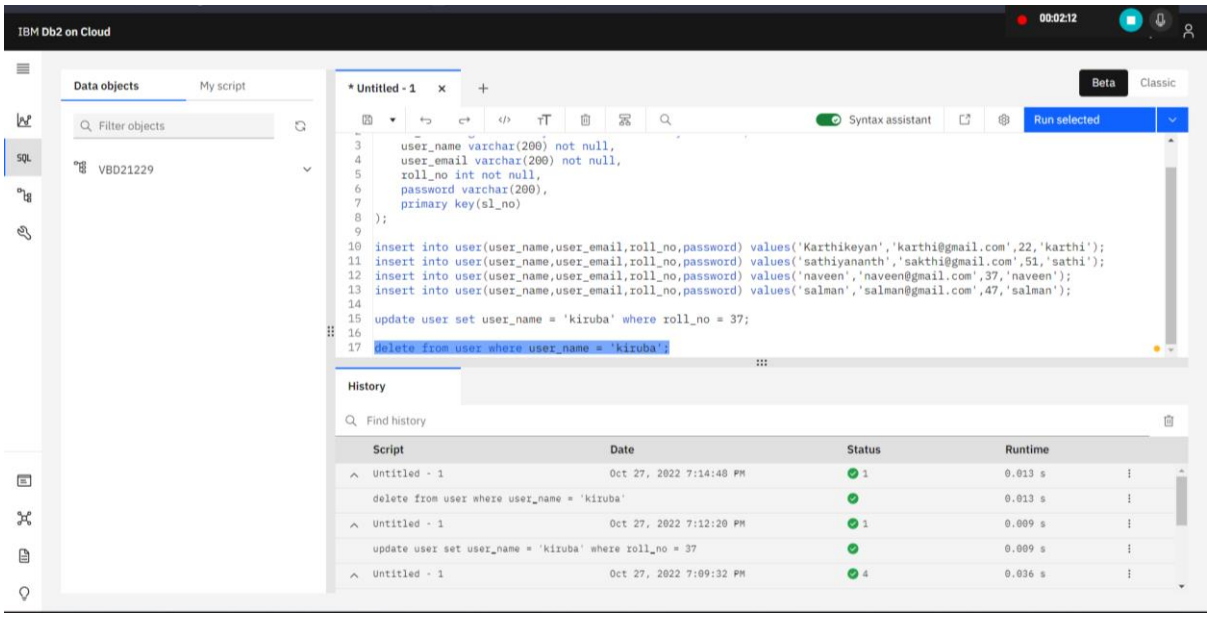
Assignment Date	12 October 2022
Student Name	Mr K Karthikeyan
Student Roll Number	621319104022
Maximum Marks	2 Marks

Question-1:

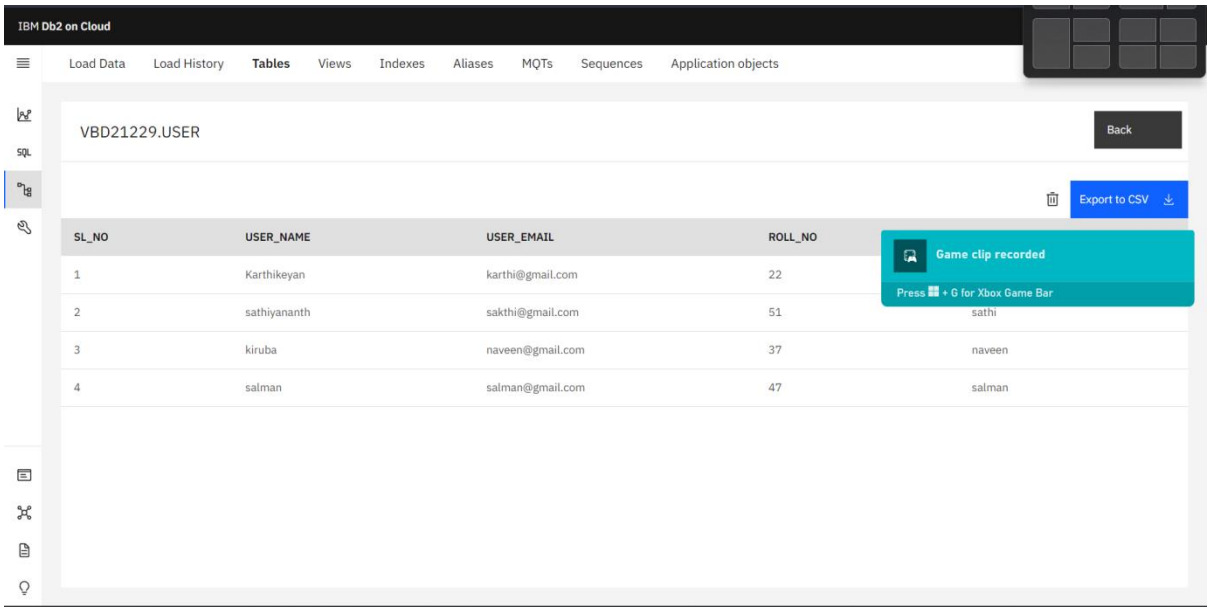
Create user table with email, username, roll number, password



Insert Values:

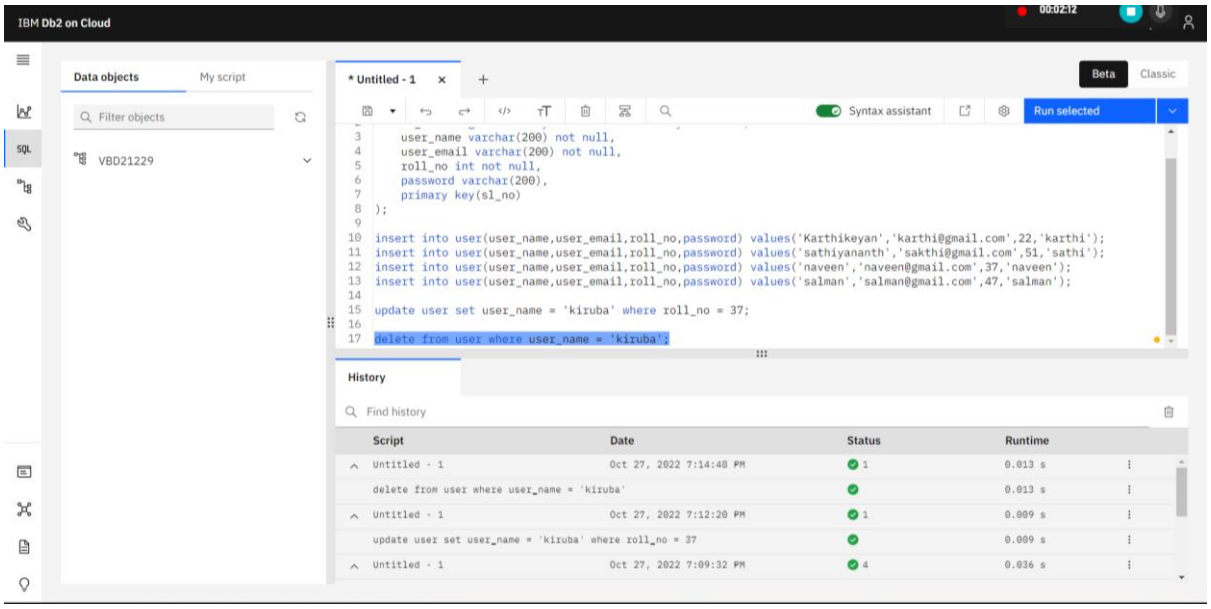


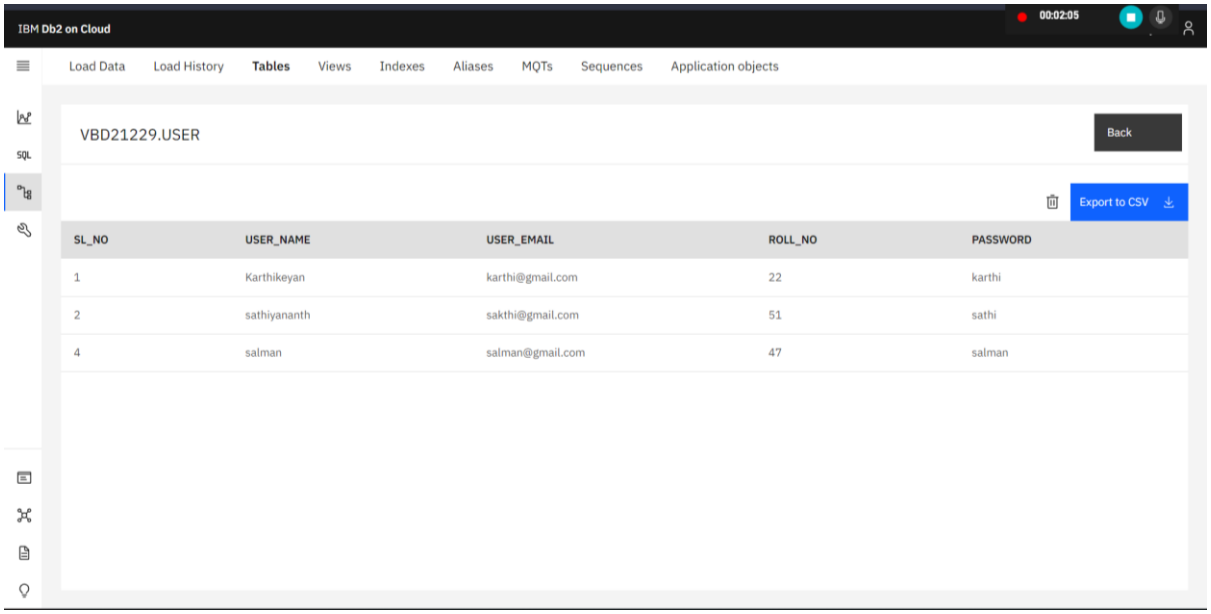
User table:



Question-2:

Perform update & delete queries with the table.





Question-3:

Connect python code to db2.

```
import ibm_db

import bcrypt

conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=1bbf73c5-d84a-4bb0-85b9ab1a4348f4a4.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud;PORT=32286;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;PROTOCOL=TCPIP;UID=vbd21229;PWD=aPuQeLB23o1pDTbz","","")
```

Question-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.

Solution:

app.py

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

```
import ibm_db

import bcrypt

conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=1bbf73c5-d84a-
4bb0-85b9-
ab1a4348f4a4.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud;PORT=32
286;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;PROTOCOL=
TCPIP;UID=vbd21229;PWD=aPuQeLB23o1pDTbz",'','')

# url_for('static', filename='style.css')

app = Flask(__name__)

app.secret_key = b'_5#y2L"F4Q8z\n\xec]/'

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

def home():

    if 'email' not in session:

        return redirect(url_for('login'))

    return render_template('home.html',name='Home')

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

def register():

  if request.method == 'POST':

    email = request.form['email']

    username = request.form['username']

    rollNo = request.form['rollNo']

    password = request.form['password']

    if not email or not username or not rollNo or not password:
```

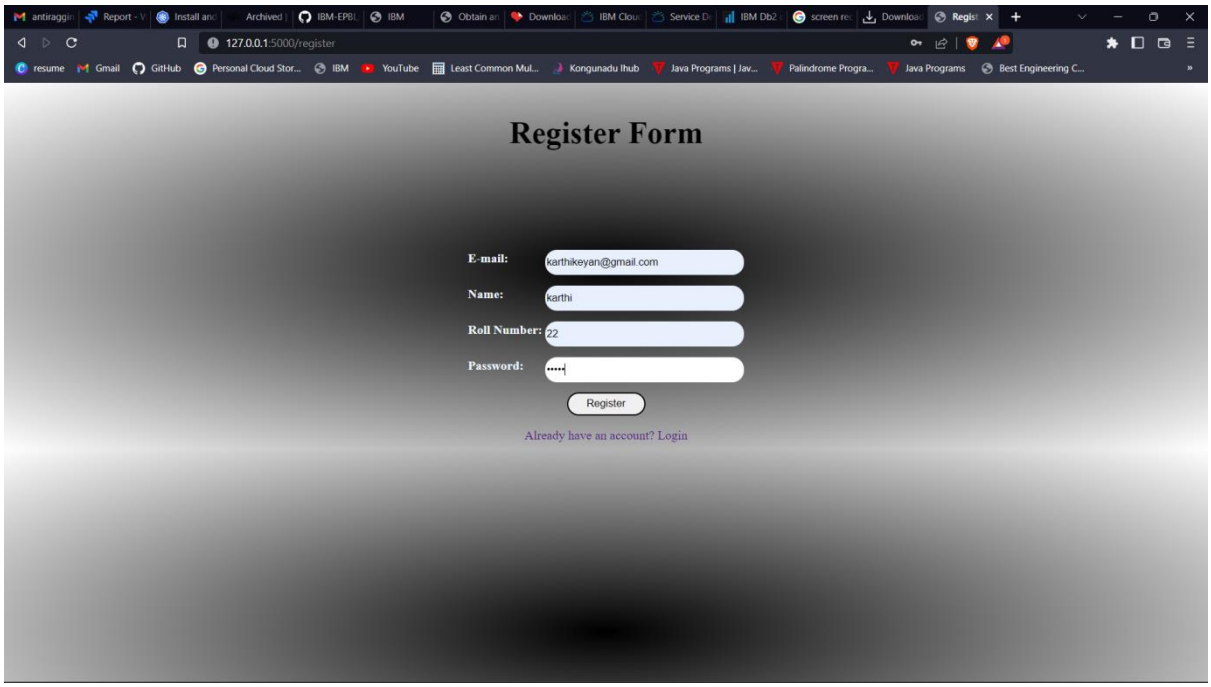
```
            return render_template('register.html',error='Please fill all fields')
        hash=bcrypt.hashpw(password.encode('utf-8'),bcrypt.gensalt())
        query = "SELECT * FROM user_details WHERE email=? OR rollNo=?"
        stmt = ibm_db.prepare(conn, query)
        ibm_db.bind_param(stmt,1,email)
        ibm_db.bind_param(stmt,2,rollNo)
        ibm_db.execute(stmt)
        isUser = ibm_db.fetch_assoc(stmt)


        if not isUser:
            insert_sql = "INSERT INTO user_details(EMAIL, USERNAME, ROLLNO,
PASSWORD) VALUES (?,?,?,?)"
            prep_stmt = ibm_db.prepare(conn, insert_sql)
            ibm_db.bind_param(prep_stmt, 1, email)
            ibm_db.bind_param(prep_stmt, 2, username)
            ibm_db.bind_param(prep_stmt, 3, rollNo)
            ibm_db.bind_param(prep_stmt, 4, hash)
            ibm_db.execute(prep_stmt)
            return render_template('register.html',success="You can login")
        else:
            return render_template('register.html',error='Invalid Credentials')
    return render_template('register.html',name='Home')
```

```
@app.route("/login",methods=['GET','POST'])
def login():
    if request.method == 'POST':
        email = request.form['email']
        password = request.form['password']

        if not email or not password:
            return render_template('login.html',error='Please fill all fields')
        query = "SELECT * FROM user_details WHERE email=?"
        stmt = ibm_db.prepare(conn, query)
        ibm_db.bind_param(stmt,1,email)
        ibm_db.execute(stmt)
        isUser = ibm_db.fetch_assoc(stmt)
        print(isUser,password)
        if not isUser:
            return render_template('login.html',error='Invalid Credentials')
        isPasswordMatch = bcrypt.checkpw(password.encode('utf-
8'),isUser['PASSWORD'].encode('utf-8'))
        if not isPasswordMatch:
            return render_template('login.html',error='Invalid Credentials')

        session['email'] = isUser['EMAIL']
```

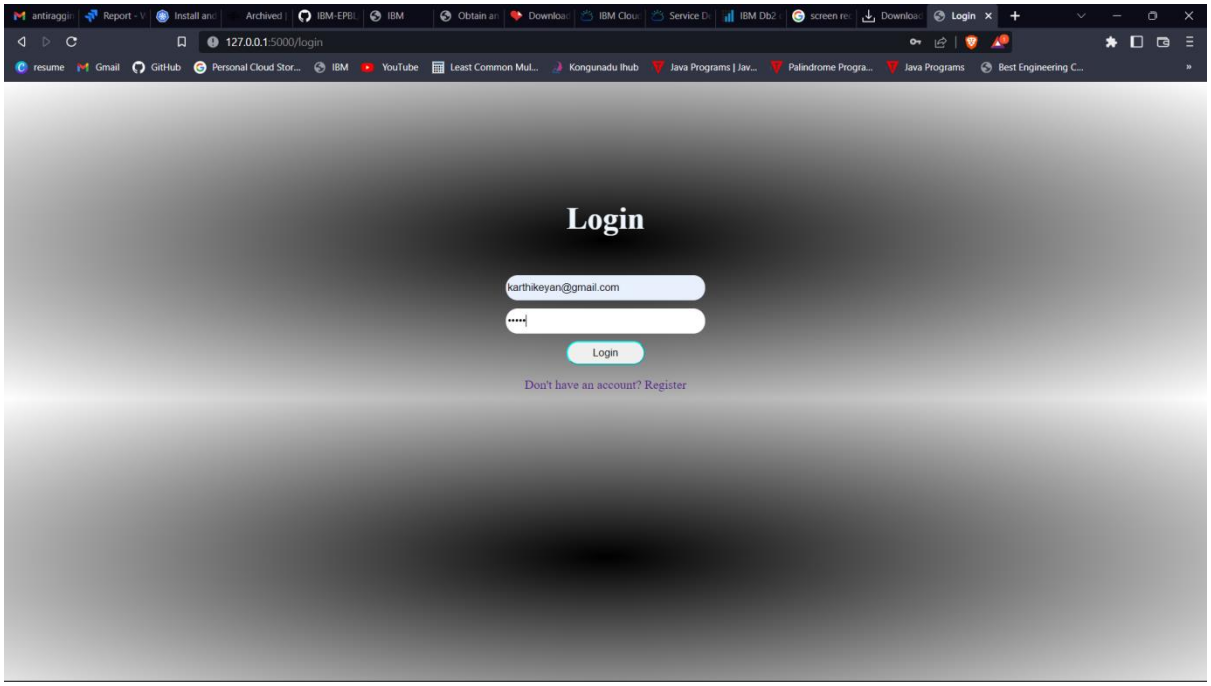
```
            return redirect(url_for('home'))
        return render_template('login.html',name='Home')
@app.route('/logout')
    def logout():
        session.pop('email', None)
        return redirect(url_for('login'))
    if __name__ == "__main__":
        app.run(debug=True)
```

Output:

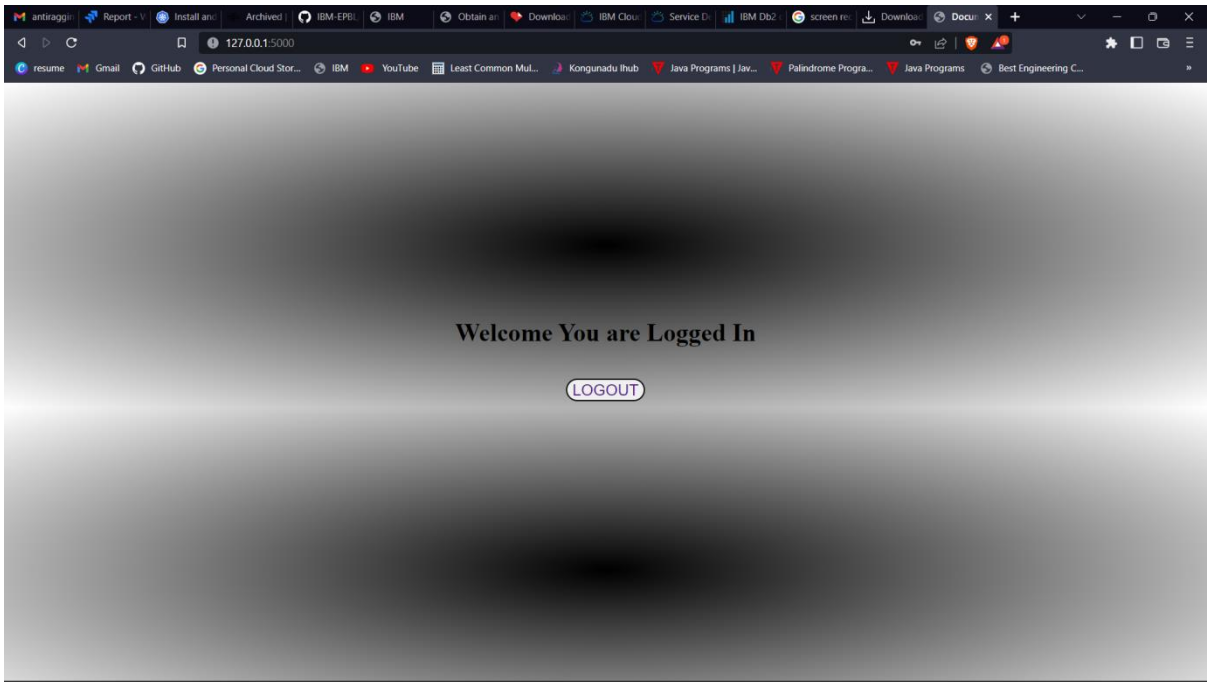
Registration Page:



Login Page:



Home Page:



Database: