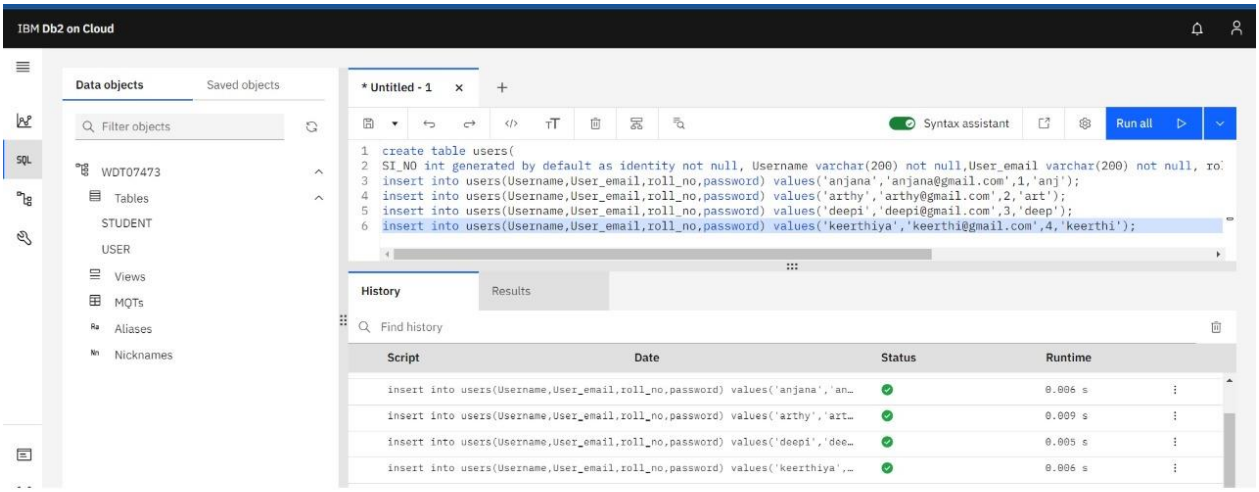
Assignment -2

DB and Flask

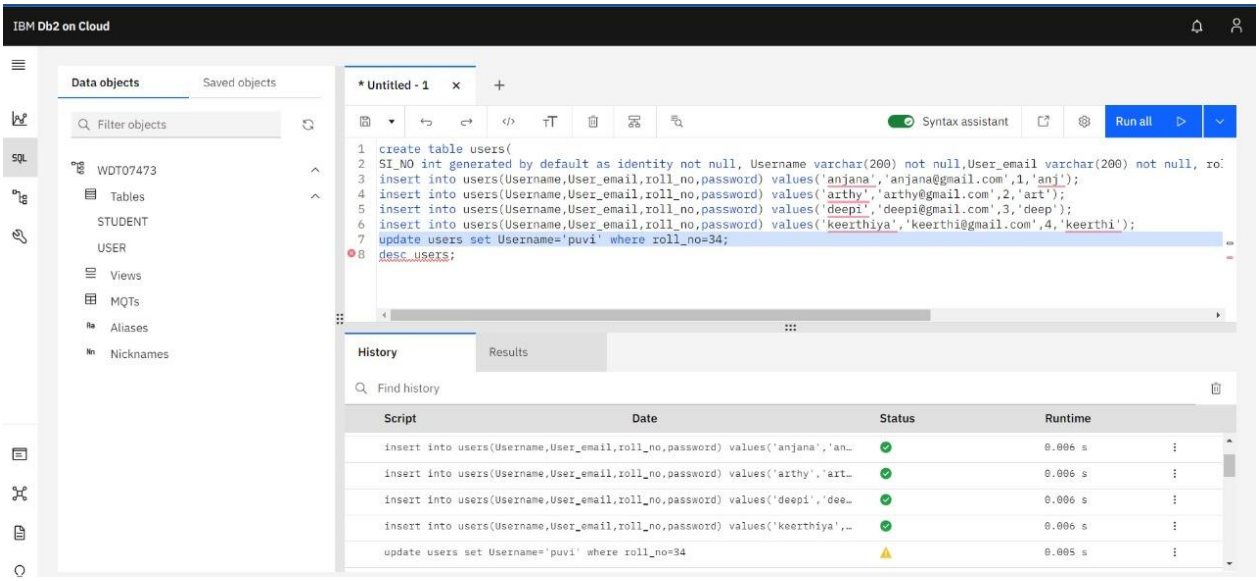
Student Name	N Arthy
Student Roll Number	621319104004
Maximum Marks	2 Marks

Question-1:

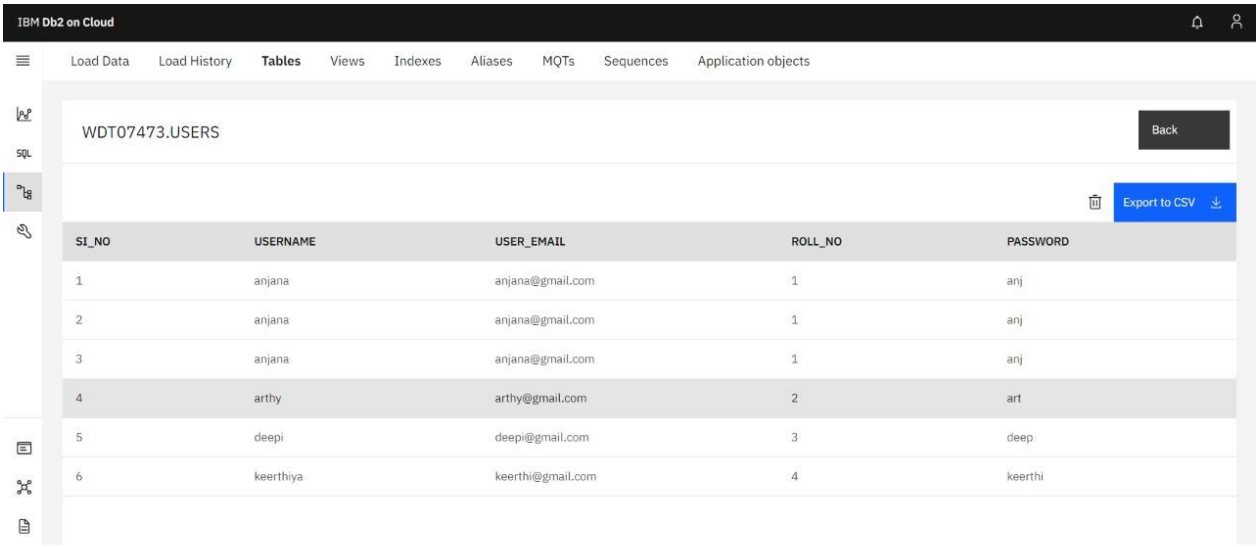
Create user table with email, username, roll number, password



Insert Values:



User table:

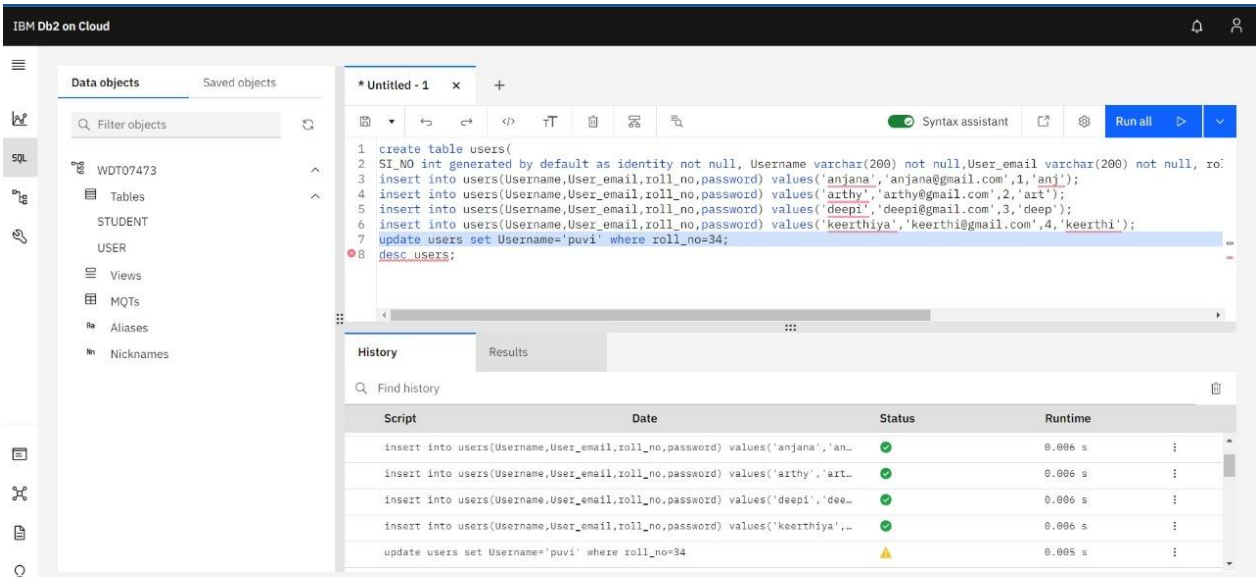


SI_NO	USERNAME	USER_EMAIL	ROLL_NO	PASSWORD
1	anjana	anjana@gmail.com	1	anj
2	anjana	anjana@gmail.com	1	anj
3	anjana	anjana@gmail.com	1	anj
4	arthy	arthy@gmail.com	2	art
5	deepi	deepi@gmail.com	3	deep
6	keerthiya	keerthi@gmail.com	4	keerthi

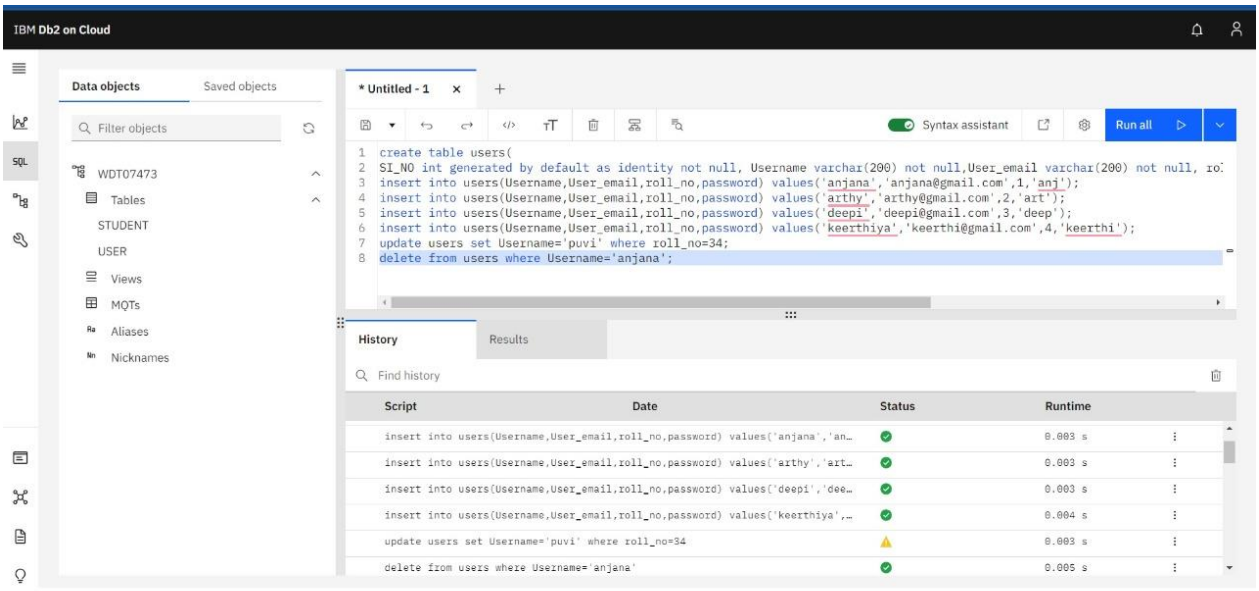
Question-2:

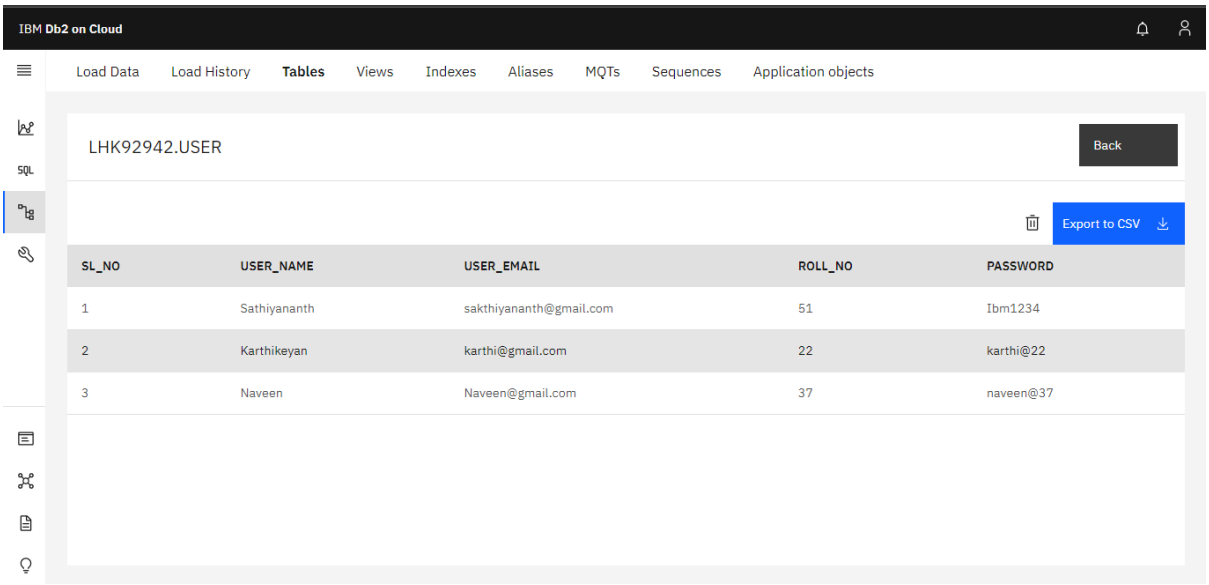
Perform update & delete queries with the table.

Update values:



Delete values:





Question-3:

Connect python code to db2.

```
import ibm_db

import bcrypt

conn =
ibm_db.connect("DATABASE=bludb;
HOSTNAME=9938aec0-8105-433e-
8bf9-
0fbb7e483086.c1ogj3sd0tgtu0lqde0
0.databases.appdomain.cloud;PORT
=32459;SECURITY=SSL;SSLServerCert
ificate=DigiCertGlobalRootCA.crt;PR
OTOCOL=TCPIP;UID=wdt07473;PWD
=khxUTQVy0OaDAOdc",'','')
```

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=blu
    db;HOSTNAME=9938aec0-8105-
    433e-8bf9-
    0fbb7e483086.c1ogj3sd0tgtu0lq
    de00.databases.appdomain.clou
    d;PORT=32459;SECURITY=SSL;SS
    LServerCertificate=DigiCertGloba
    lRootCA.crt;PROTOCOL=TCPIP;UI
    D=wdt07473;PWD=khxUTQVy0O
    aDAOdc","","")

# 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('index.html',na
    me='Home')

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

   if not email or not name or not
    rollNo or not password:
     return
    render_template('register.html',
    error='Please fill all fields')
```

```
 hash=bcrypt.hashpw(password.e
 ncode('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,emai
 l)

 ibm_db.bind_param(stmt,2,rollN
 o)
ibm_db.execute(stmt)
isUser =
 ibm_db.fetch_assoc(stmt)

if not isUser:
  insert_sql = "INSERT INTO
 user_details(EMAIL, NAME,
 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, name)
```

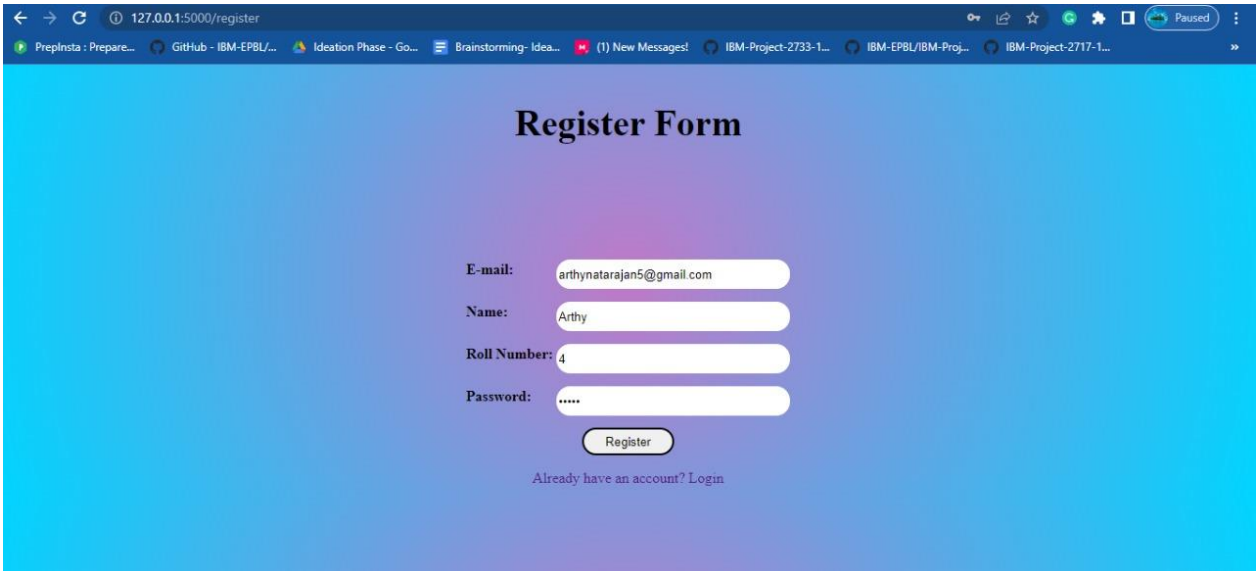
```
      ibm_db.bind_param(prep_stmt,
     3, rollNo)
      ibm_db.bind_param(prep_stmt,
     4, hash)
      ibm_db.execute(prep_stmt)
      return
     render_template('register.html',s
     uccess="You can login")
    else:
      return
     render_template('register.html',
     error='Invalid Credentials')

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

@app.route("/login",methods=['GE
     T','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',err
     or='Please fill all fields')
      query = "SELECT * FROM
```

```
user_details WHERE email=?"
 stmt = ibm_db.prepare(conn,
query)

ibm_db.bind_param(stmt,1,emai
l)
 ibm_db.execute(stmt)
 isUser =
ibm_db.fetch_assoc(stmt)
 print(isUser,password)

 if not isUser:
   return
render_template('login.html',err
or='Invalid Credentials')

 isPasswordMatch =
bcrypt.checkpw(password.encod
e('utf-
8'),isUser['PASSWORD'].encode('
utf-8'))
 if not isPasswordMatch:
   return
render_template('login.html',err
or='Invalid Credentials')

 session['email'] = isUser['EMAIL']
 return redirect(url_for('home'))

return
```

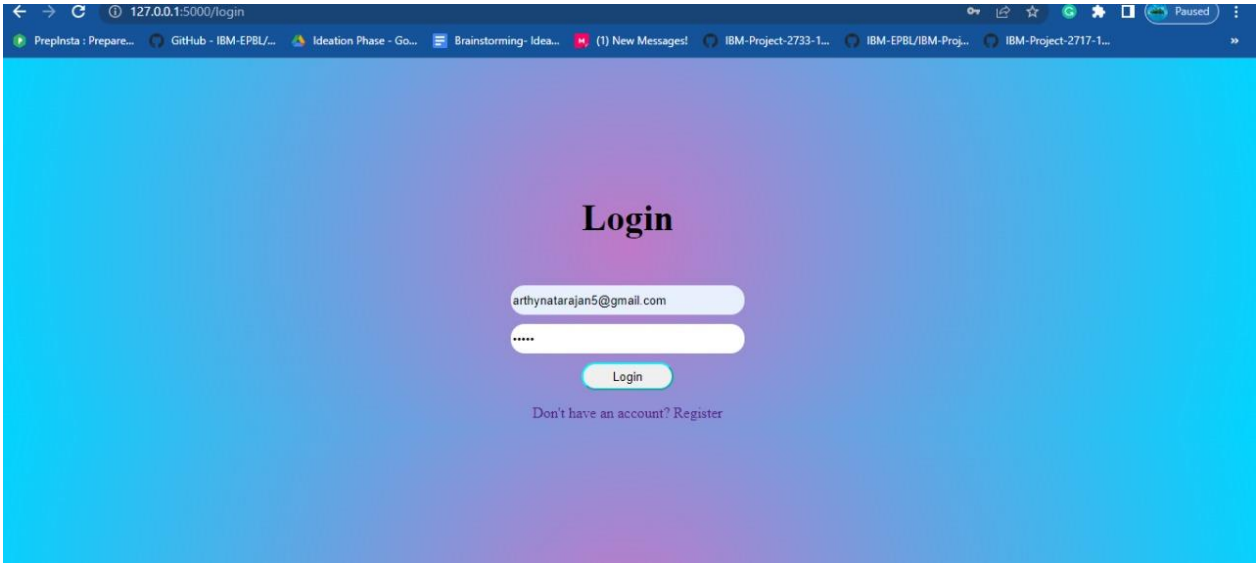
```
    render_template('login.html',na
    me='Home')
@app.route('/logout')
def logout():
   session.pop('email', None)
   return redirect(url_for('login'))
if _name_ == "_main_":
   app.run(debug=True)
```

Output:

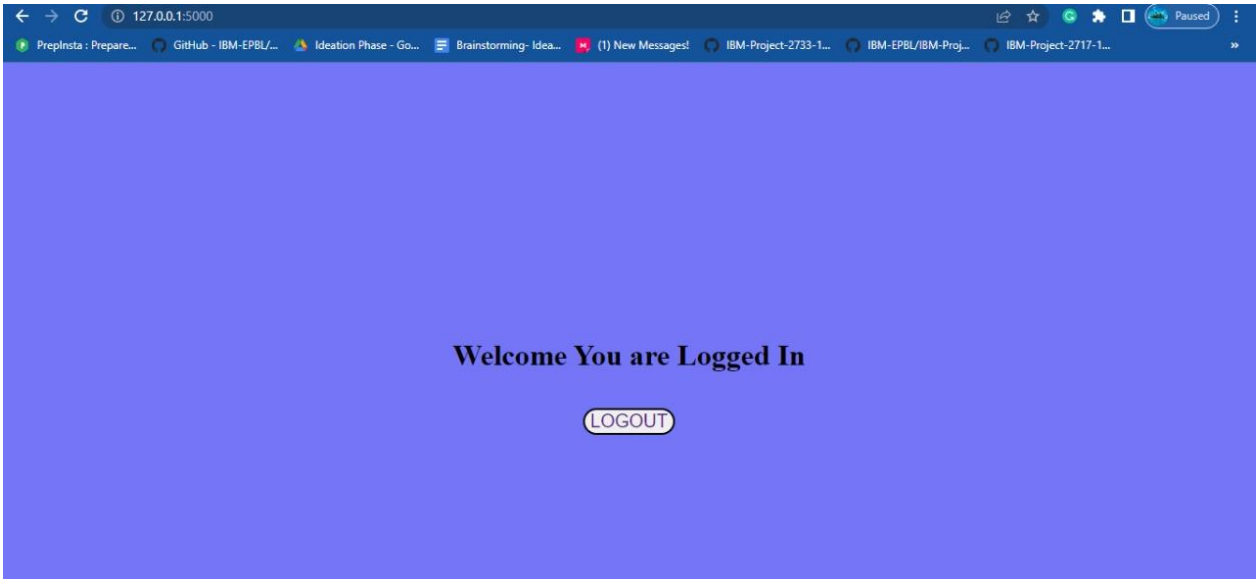
Registration Page:



Login Page:



Home Page:



Database: