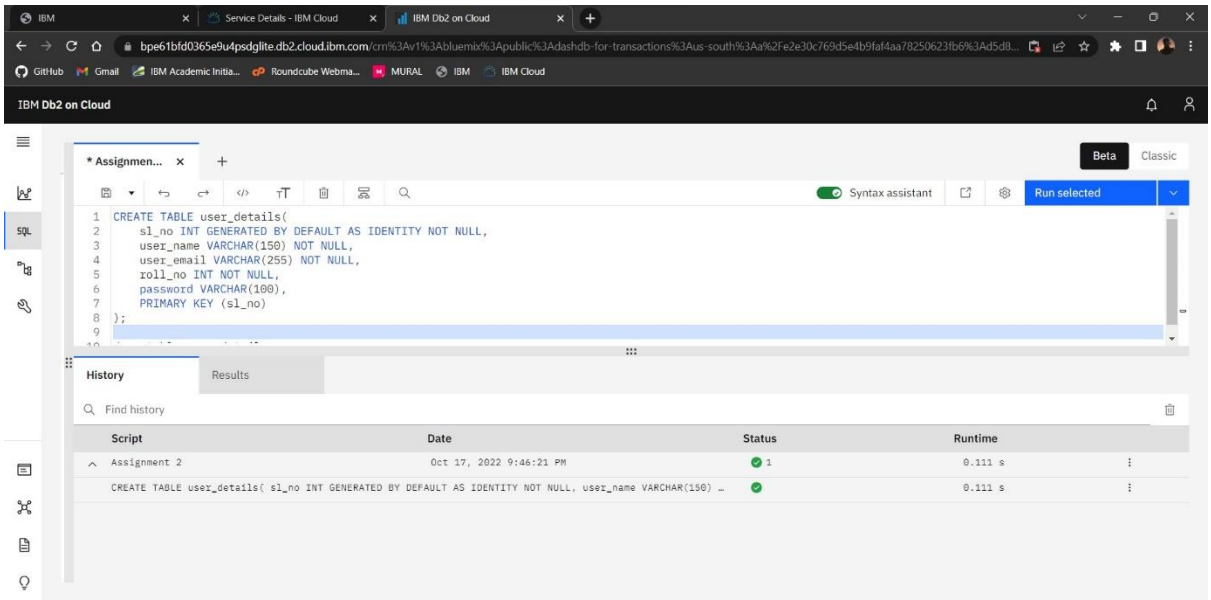
# ASSIGNMENT 2

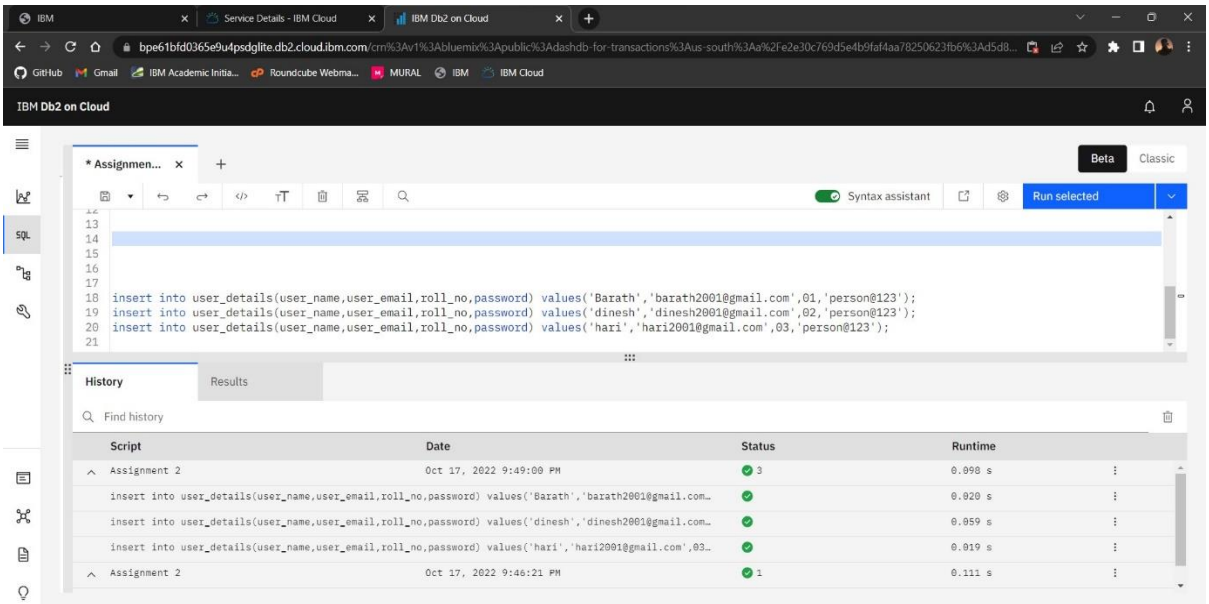
| | |
|---|---|
| Date | 12 October 2022 |
| Student Name | Barathkumar P |
| Student Roll no | 621319106008 |
| Maximum Marks | 2 Marks |

## 1. Create User table with email, username, roll number, password.

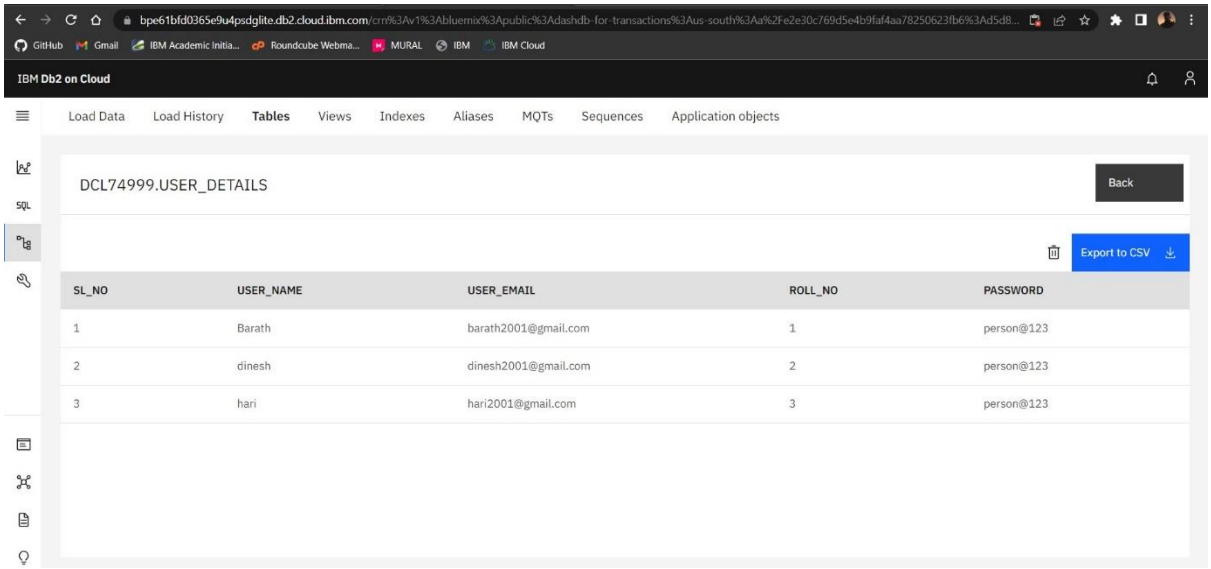
Create user table:




Insert details:

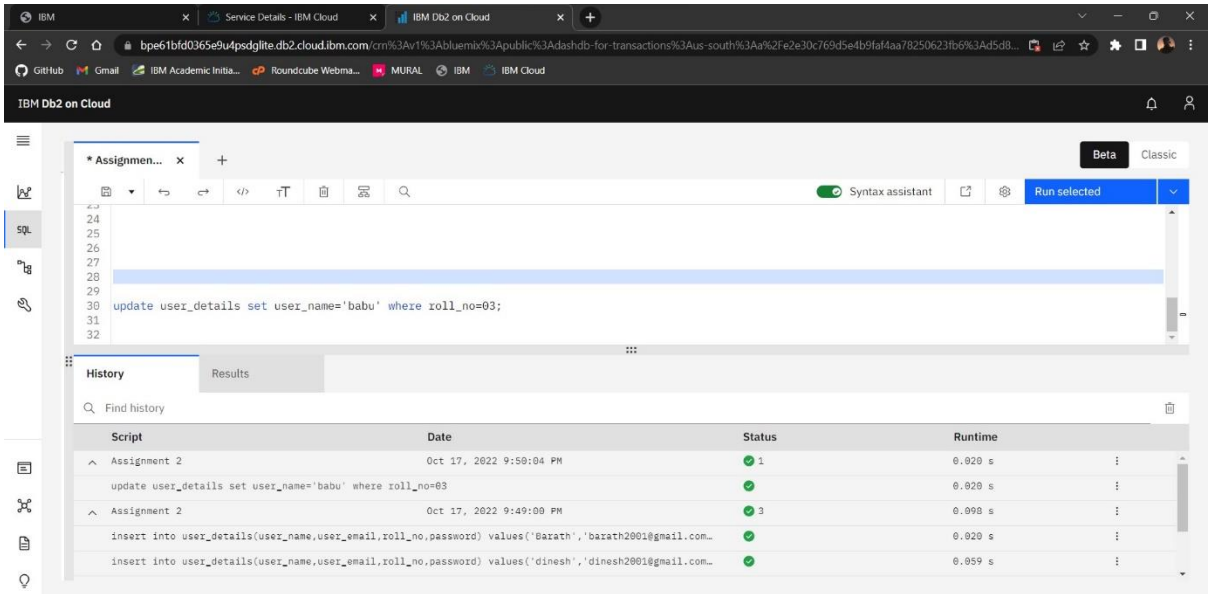


User Table:

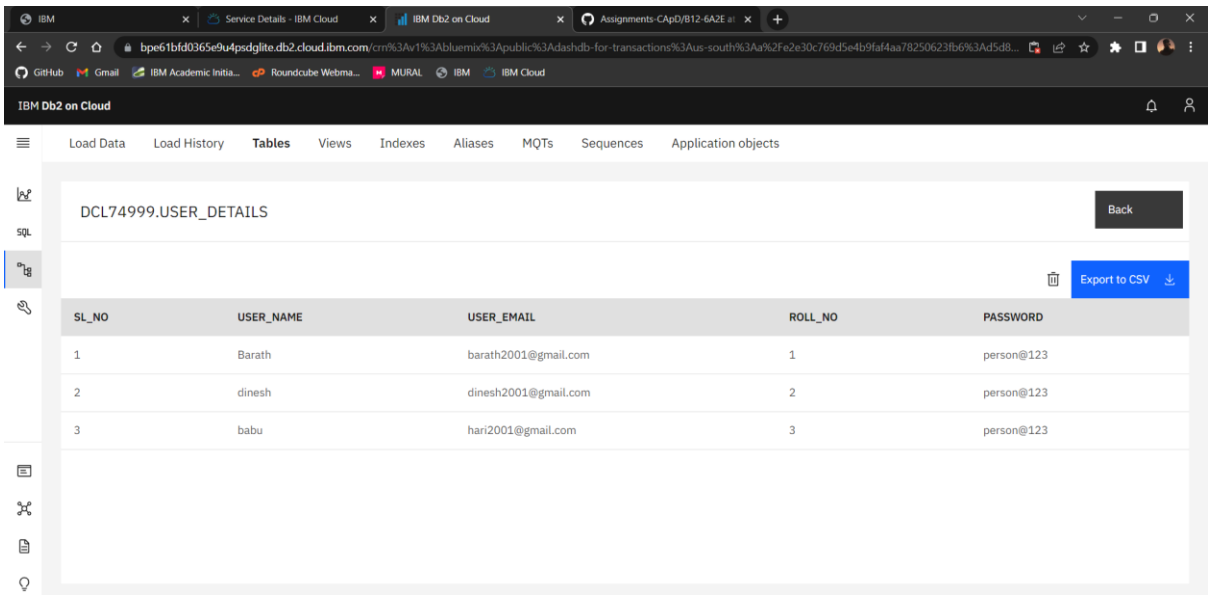



## 2. Perform UPDATE, DELETE Queries with user table

Update:



Updated Table:




| SL_NO | USER_NAME | USER_EMAIL | ROLL_NO | PASSWORD |
|---|---|---|---|---|
| 1 | Barath | barath2001@gmail.com | 1 | person@123 |
| 2 | dinesh | dinesh2001@gmail.com | 2 | person@123 |
| 3 | babu | hari2001@gmail.com | 3 | person@123 |

Delete:

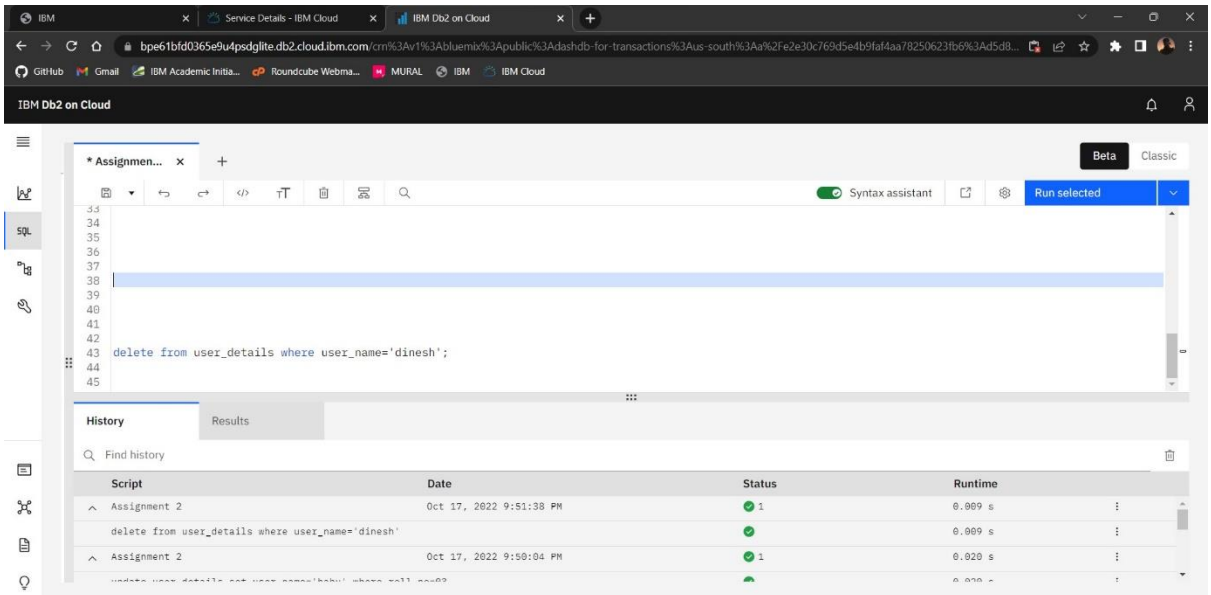
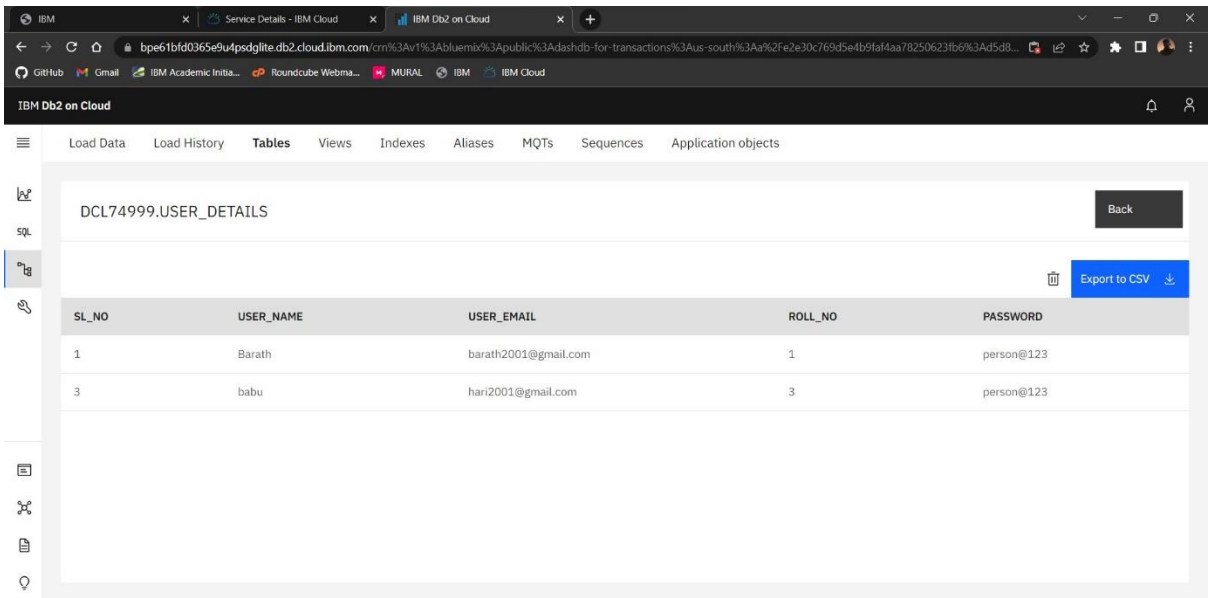


Table after performed delete query




**3. Connect python code to db2.**

```
import ibm_db
import bcrypt
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=31321;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;PROTOCOL=TCPIP;UID=dcl74999;PWD=Px6tqEmXL9AcQigs",'','')
```

**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 with username and password. If the user is valid show the welcome page**

App.py

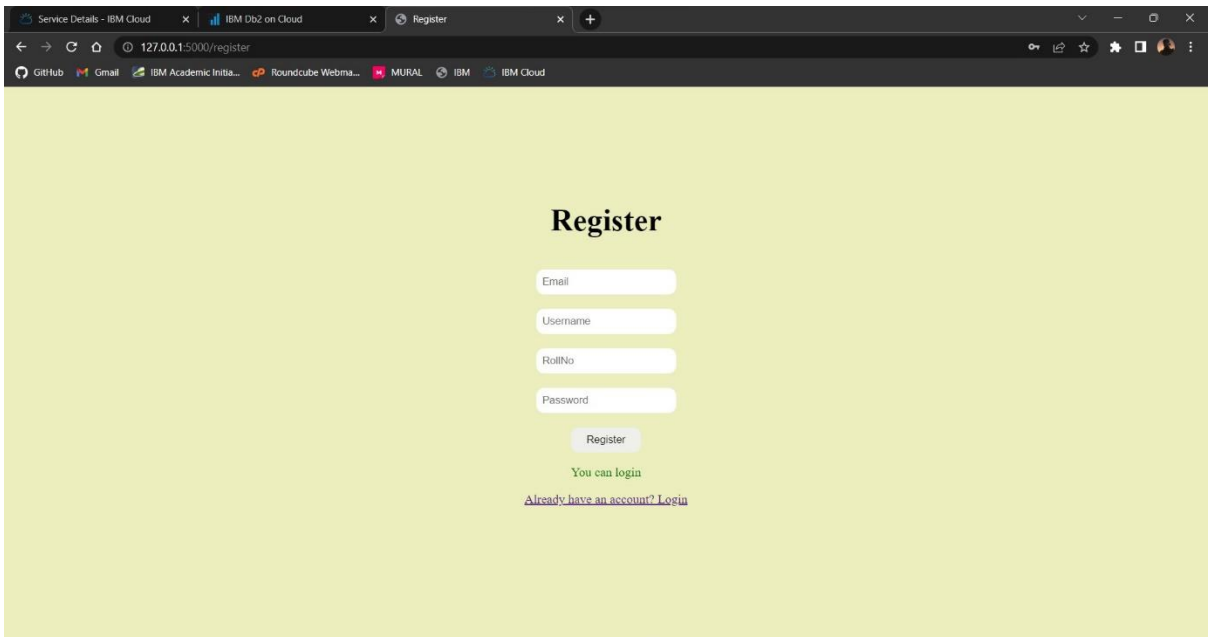
```
from flask import Flask, render_template, request, redirect, url_for, session

import ibm_db
import bcrypt
```

```
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=ba99a9e6-d59e-4883-
8fc0-
d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=31321;
SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;PROTOCOL=TCP
IP;UID=dcl74999;PWD=Px6tqEmXL9AcQigs",'','')


# 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 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(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)
```

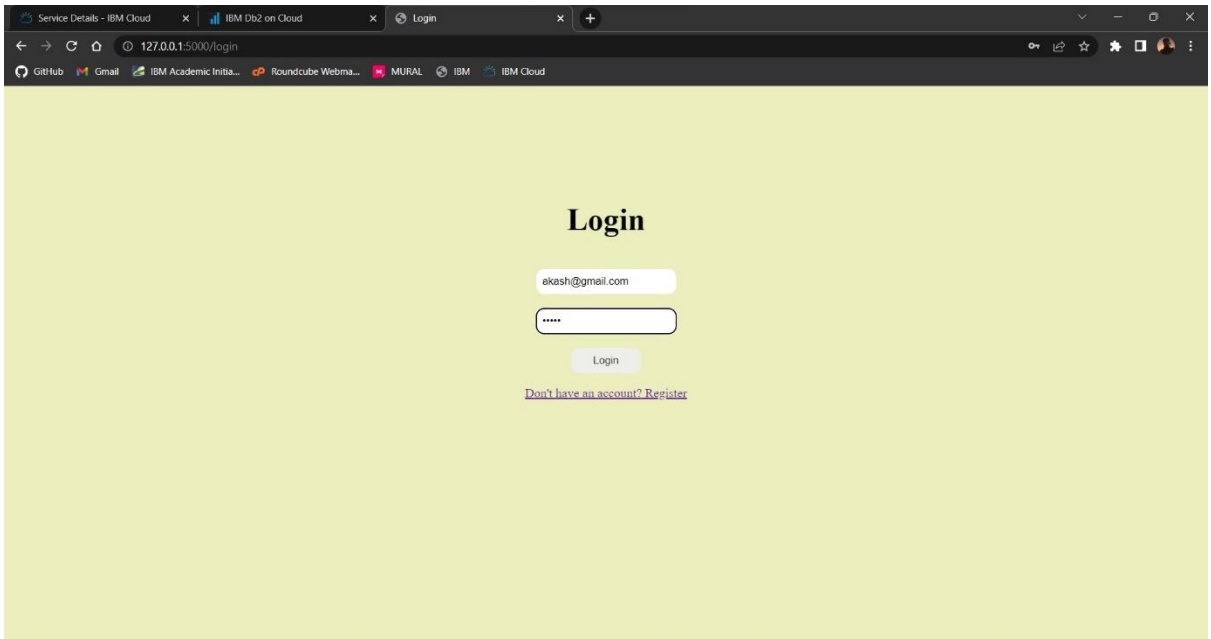
```
      ibm_db.bind_param(prep_stmt, 3, rollNo)
      ibm_db.bind_param(prep_stmt, 4, hash)
      ibm_db.execute(prep_stmt)
      return render_template('register.html',success="You can login")
    else:
      return render_template('register.html',error='Invalid Credentials')

  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 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:**

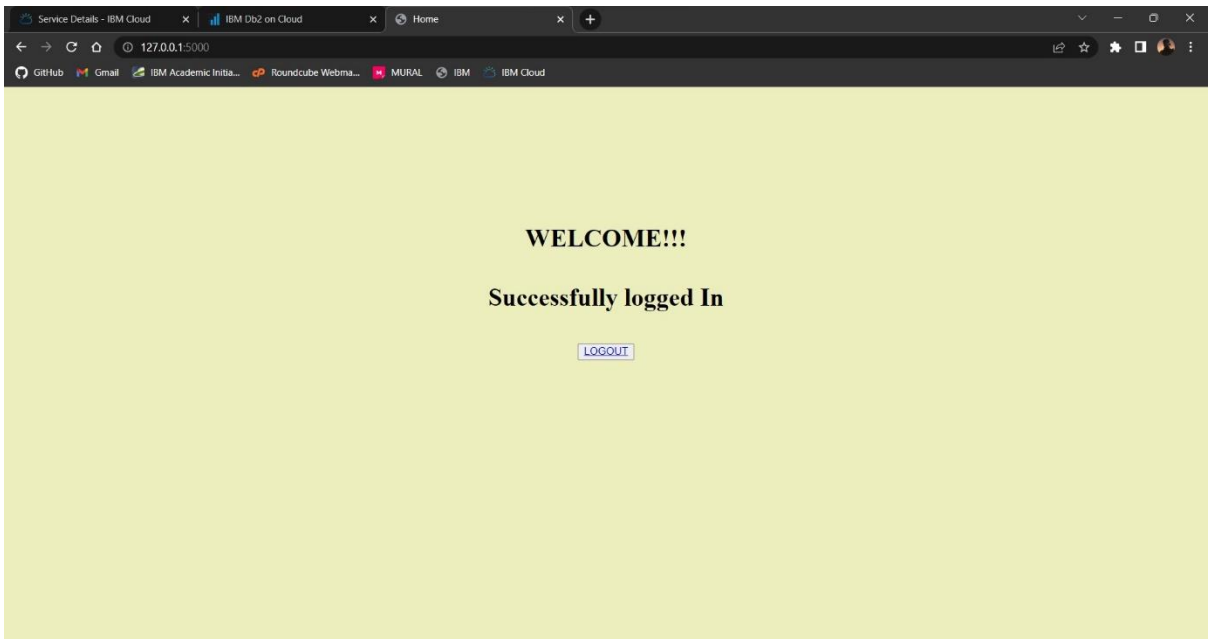
**Registering:**




**Logging in:**



## Home page:




## Database: