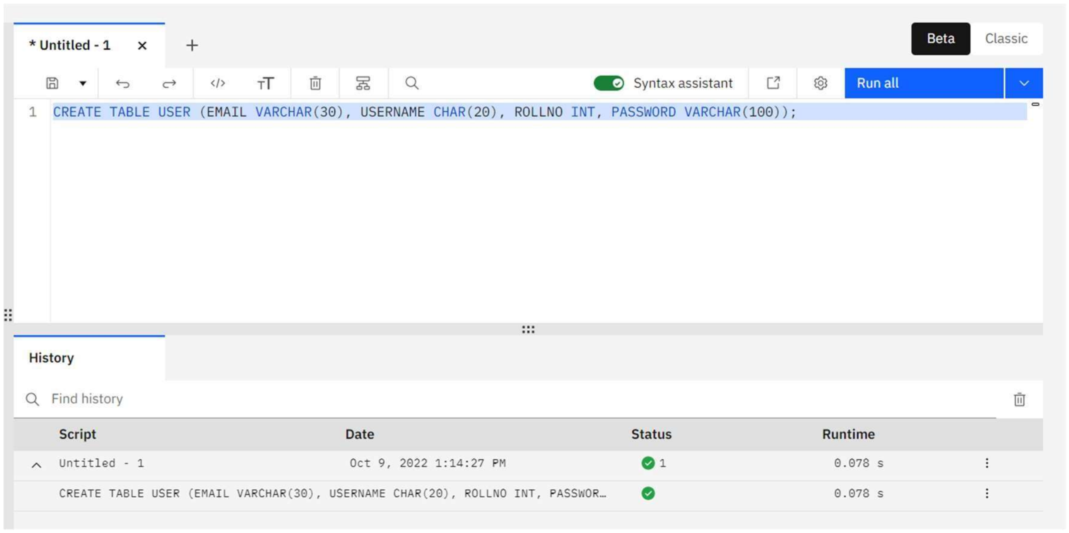
Assignment -2

Connect with Database Assignment

Assignment Date	09 October 2022
Student Name	Kishorekumar P
Student Roll Number	621319104027
Maximum Marks	2 Marks

Question-1:

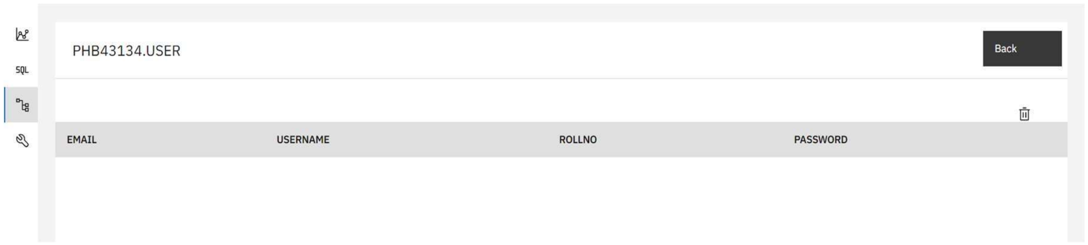
1. Create User table with user with email , username, roll number, password.



```
CREATE TABLE USER (EMAIL VARCHAR(30), USERNAME CHAR(20), ROLLNO INT, PASSWORD VARCHAR(100));
```

History

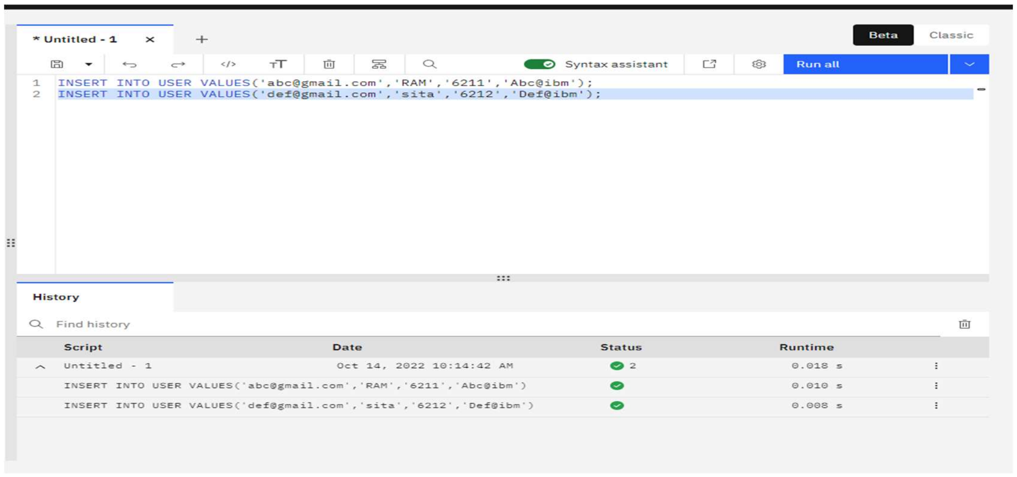
Script	Date	Status	Runtime
Untitled - 1	Oct 9, 2022 1:14:27 PM	✓ 1	0.078 s
CREATE TABLE USER (EMAIL VARCHAR(30), USERNAME CHAR(20), ROLLNO INT, PASSWOR…		✓	0.078 s



PHB43134.USER

EMAIL	USERNAME	ROLLNO	PASSWORD

2. Perform UPDATE, DELETE Queries with User table

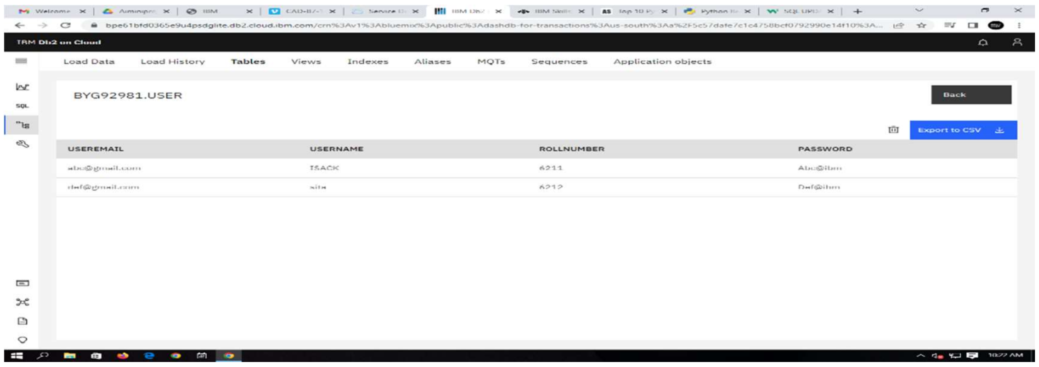


```
INSERT INTO USER VALUES('abc@gmail.com','RAM','6211','Abc@ibm');
INSERT INTO USER VALUES('def@gmail.com','sita','6212','Def@ibm');
```

History

Script	Date	Status	Runtime
Untitled - 1	Oct 14, 2022 10:14:42 AM	✓ 2	0.018 s
INSERT INTO USER VALUES('abc@gmail.com','RAM','6211','Abc@ibm')		✓	0.010 s
INSERT INTO USER VALUES('def@gmail.com','sita','6212','Def@ibm')		✓	0.008 s

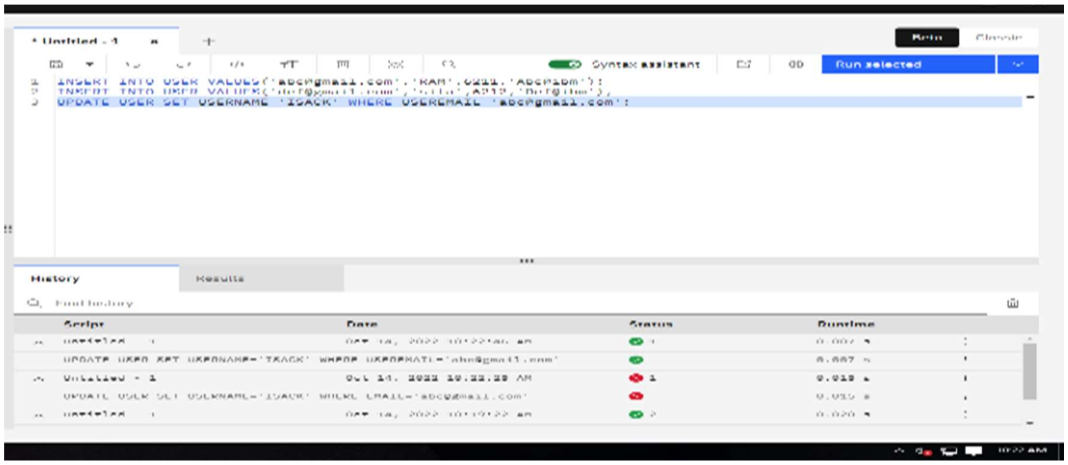
Table View:



BYG92981.USER

USEREMAIL	USERNAME	ROLLNUMBER	PASSWORD
abc@gmail.com	ISACK	6211	Abc@ibm
def@gmail.com	sita	6212	Def@ibm

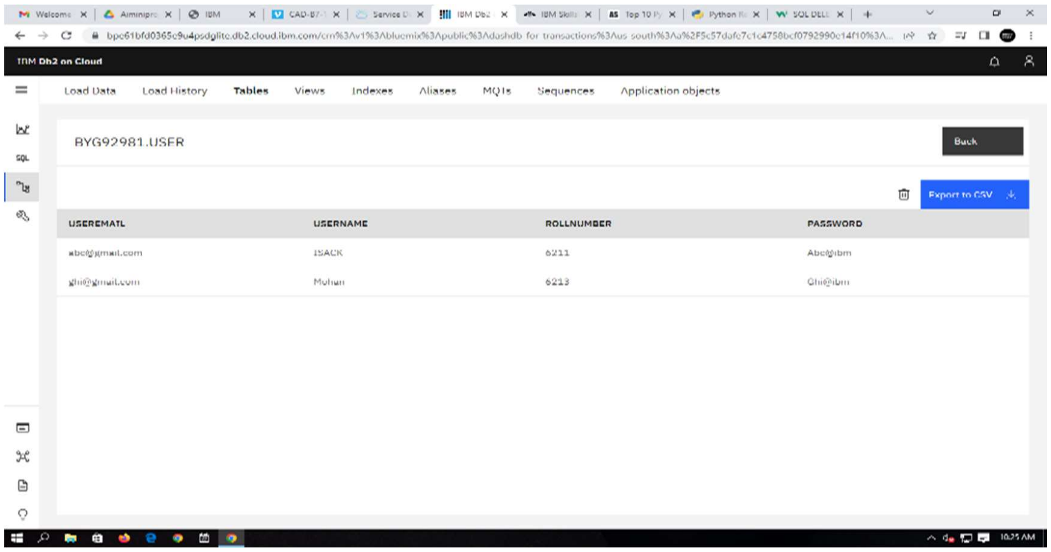
UPDATE:



```
INSERT INTO USER VALUES('abc@gmail.com','RAM',6211,'Abc@ibm');
INSERT INTO USER VALUES('def@gmail.com','sita',6212,'Def@ibm');
UPDATE USER SET USERNAME 'ISACK' WHERE USEREMAIL 'abc@gmail.com';
```

```
Script                                                  Date                        Status   Runtime
Untitled - 1                                            Oct 14, 2022 10:22:46 AM    1        0.007 s
UPDATE USER SET USERNAME='ISACK' WHERE USEREMAIL='abc@gmail.com'                             0.007 s
Untitled - 1                                            Oct 14, 2022 10:22:28 AM    1        0.019 s
UPDATE USER SET USERNAME='ISACK' WHERE EMAIL='abc@gmail.com'                                 0.015 s
Untitled - 1                                            Oct 14, 2022 10:19:22 AM    2        0.020 s
```

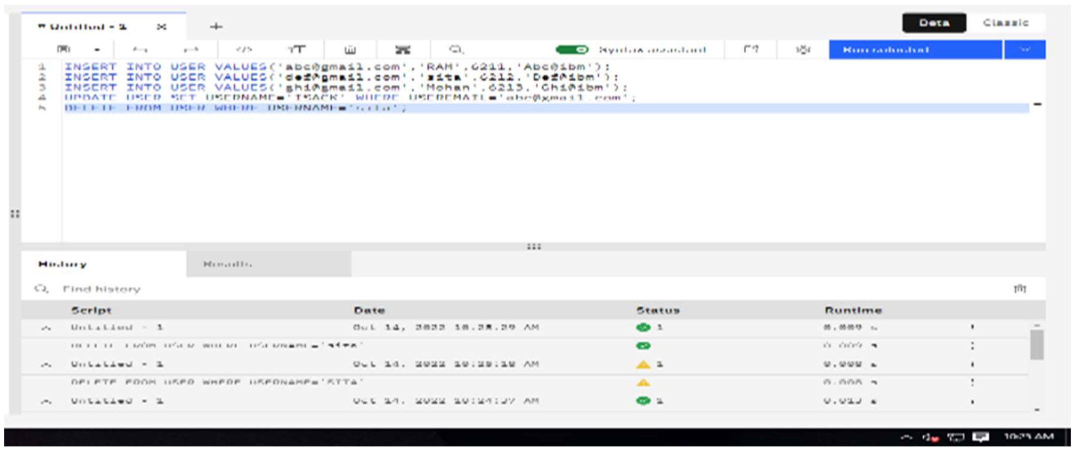
Table View:



BYG92981.USER

USEREMAIL	USERNAME	ROLLNUMBER	PASSWORD
abc@gmail.com	ISACK	6211	Abc@ibm
ghi@gmail.com	Mohan	6213	Ghi@ibm

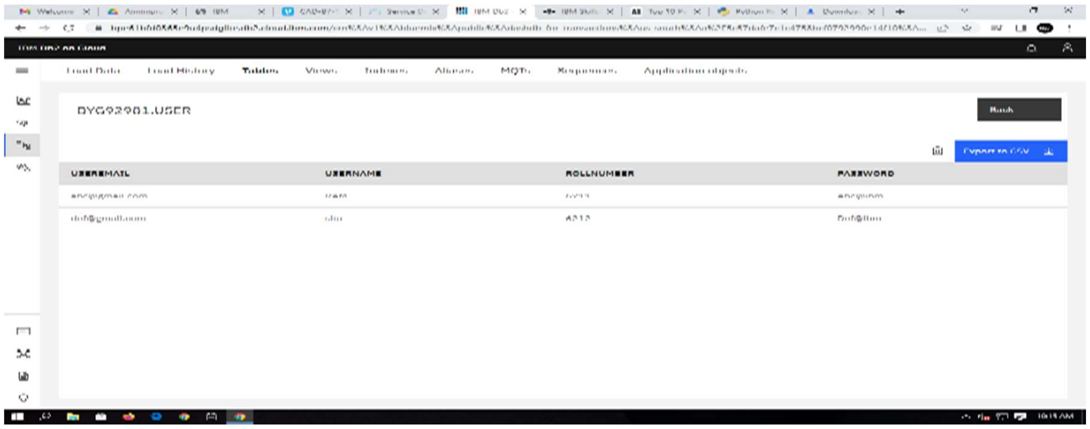
DELETE:



```
INSERT INTO USER VALUES('abc@gmail.com','RAM',6211,'Abc@ibm');
INSERT INTO USER VALUES('def@gmail.com','sita',6212,'Def@ibm');
INSERT INTO USER VALUES('ghi@gmail.com','Mohan',6213,'Ghi@ibm');
UPDATE USER SET USERNAME='ISACK' WHERE USEREMAIL='abc@gmail.com';
DELETE FROM USER WHERE USERNAME='sita';
```

Script	Date	Status	Runtime
Untitled - 1	Oct 14, 2022 10:28:29 AM	● 1	0.009 s
DELETE FROM USER WHERE USERNAME='sita'		●	0.002 s
Untitled - 1	Oct 14, 2022 10:28:18 AM	⚠ 1	0.008 s
DELETE FROM USER WHERE USERNAME='SITA'		⚠	0.008 s
Untitled - 1	Oct 14, 2022 10:24:37 AM	● 1	0.012 s

TABLE View:



DYG92981.USER

USEREMAIL	USERNAME	ROLLNUMBER	PASSWORD
abc@gmail.com	ram	6211	abc@ibm
def@gmail.com	sita	6212	Def@ibm

3. Connect python with db2.

Solution:

```
import ibm_db
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME= 1bbf73c5-d84a-4bb0-85b9-ab1a4348f4a4.c3n41cmd0
nqnrk39u98g.databases.appdomain.cloud;PORT=32286;SECURITY=SSL;SSLS erverCertificate=DigiCertGl
obalRootCA.crt;PROTOCOL=TCPIP;UID= byg92981;PWD=” 9jZpv8EpbeEMaB6i",'','')
```

4. Create a flask app with the registration page. Login page and the 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 so the welcome page.

Solution:

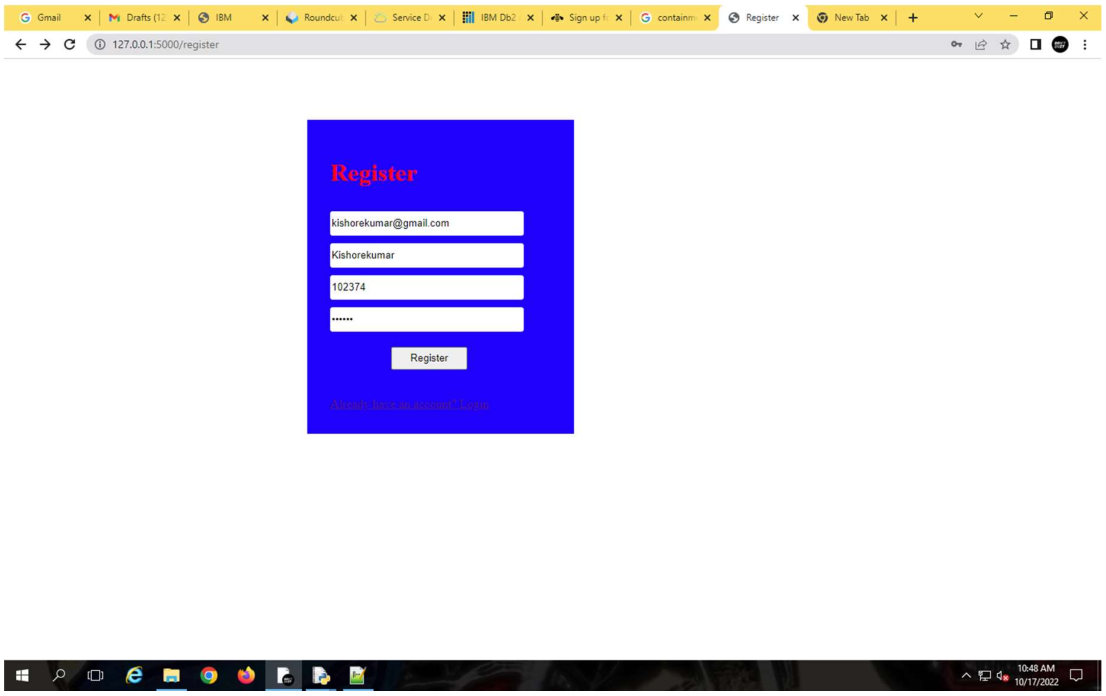
app.py

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

import ibm_db
import bcrypt
conn =
ibm_db.connect("DATABASE=bludb;HOSTNAME=;PORT=;SECURITY=SSL;SSLServerCertificate=DigiCer
tGlobalRootCA.crt;UID=;PWD=",'','')

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

app = Flask(_name_)
app.secret_key = 'C21FGSBAPOK43K5VSIDFB2'

@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(username,email,PASSWORD,rollNo) VALUES (?,?,?,?)"
      prep_stmt = ibm_db.prepare(conn, insert_sql)
      ibm_db.bind_param(prep_stmt, 1, username)
      ibm_db.bind_param(prep_stmt, 2, email)
      ibm_db.bind_param(prep_stmt, 3, hash)
      ibm_db.bind_param(prep_stmt, 4, rollNo)
      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)
```

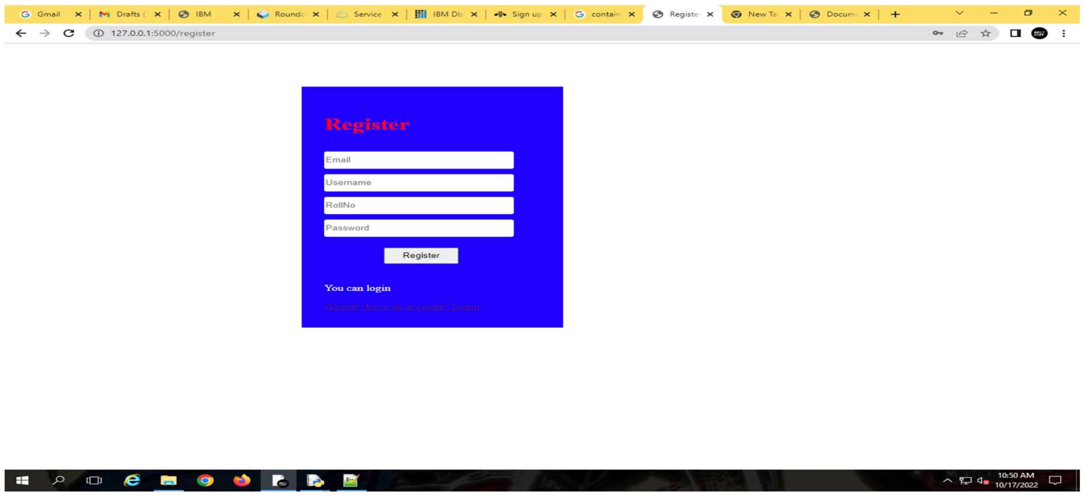
```
   if not isUser:
    return render_template('login.html',error='Invalid Credentials')

   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'))
```

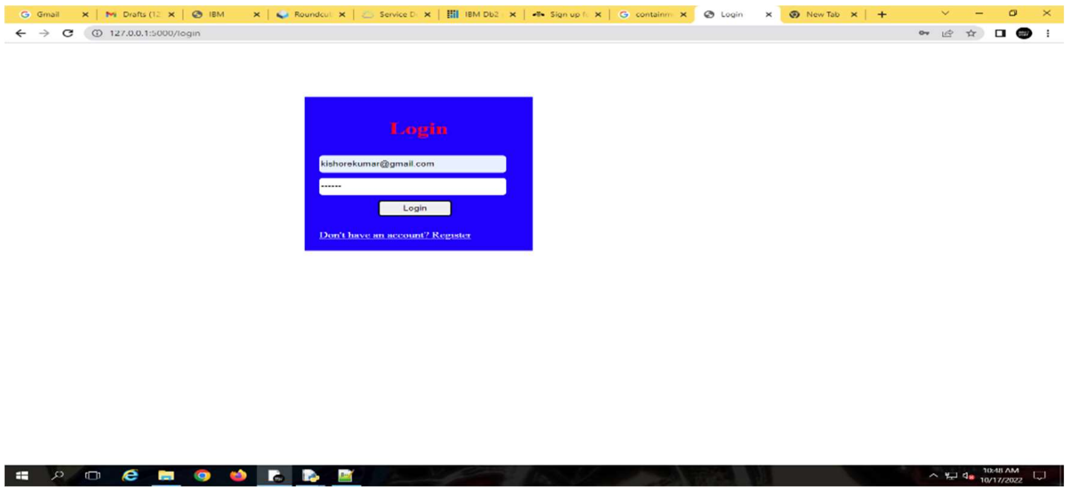
OUTPUT:

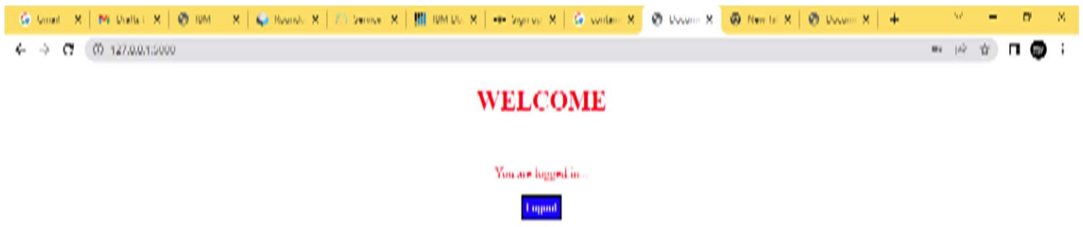


kishorekumar@gmail.com
Kishorekumar
102374
••••••

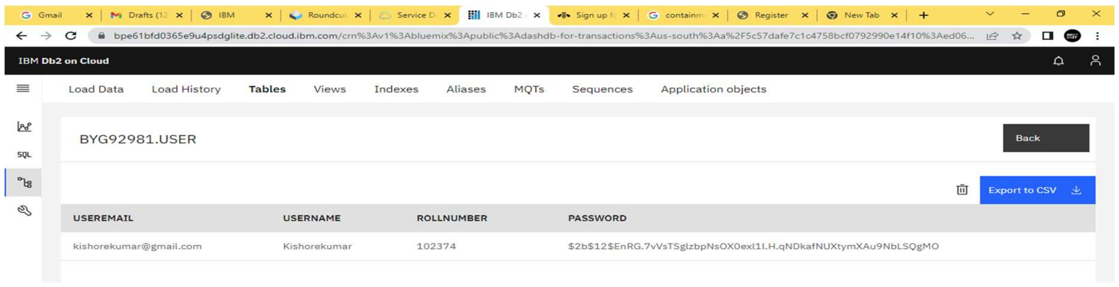
Already have an account? Login







Database:


bpe61bfd0365e9u4psdglite.db2.cloud.ibm.com/crn%3Av1%3Abluemix%3Apublic%3Adashdb-for-transactions%3Aus-south%3Aa%2F5c57dafe7c1c4758bcf0792990e14f10%3Aed06...
IBM Db2 on Cloud
Load Data
Load History
Tables
Views
Indexes
Aliases
MQTs
Sequences
Application objects
SQL
Back
Export to CSV

BYG92981.USER

USEREMAIL	USERNAME	ROLLNUMBER	PASSWORD
kishorekumar@gmail.com	Kishorekumar	102374	$2b$12$EnRG.7vVsTSglzbpNsOX0exl1l.H.qNDkafNUXtymXAu9NbLSQgMO