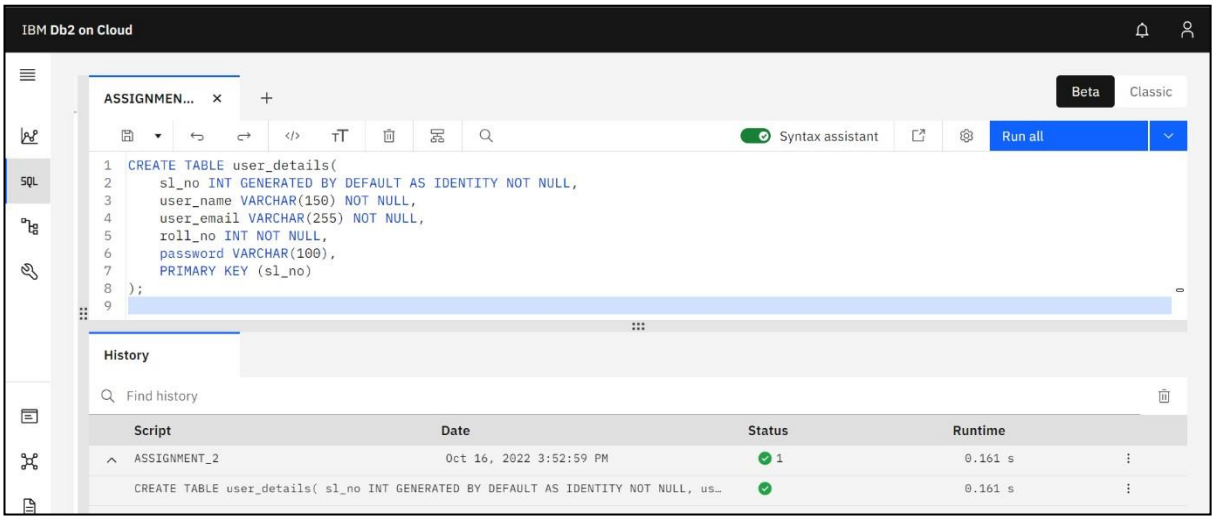
ASSIGNMENT 2

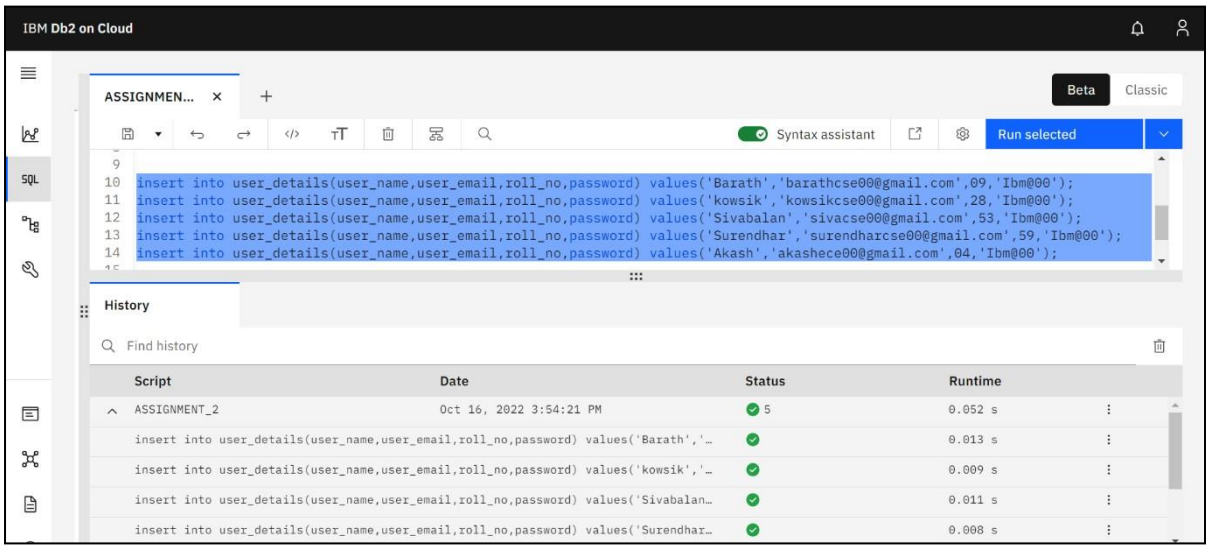
Database connection

Date	12 October 2022
Student Name	Kowsik v
Student Roll no	621319104028
Maximum Marks	2 Marks

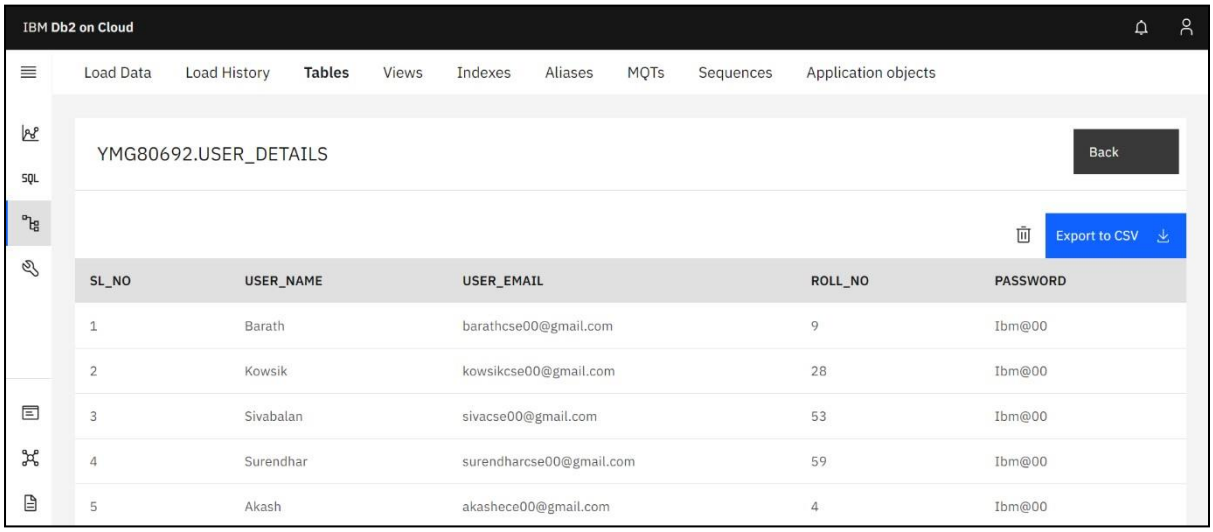
1. Create user table with email, username, roll number, password



Inserting values :

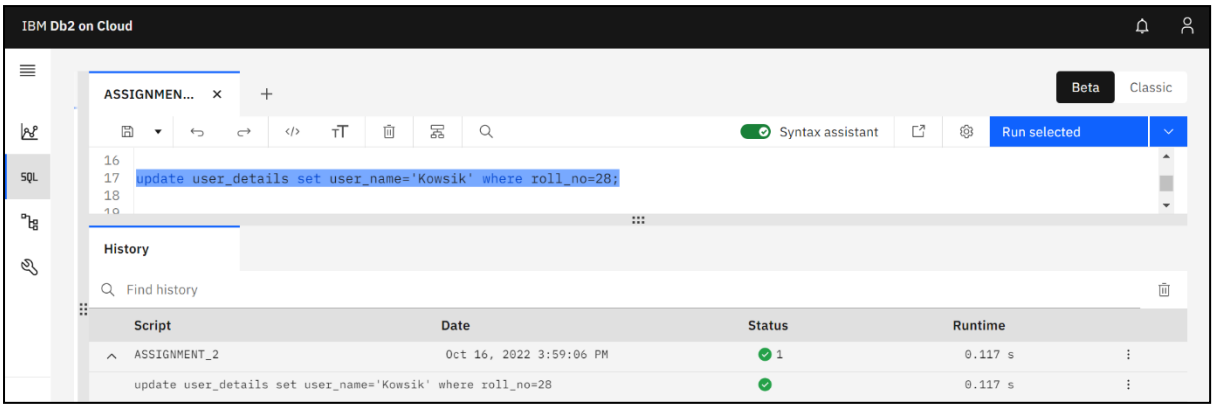


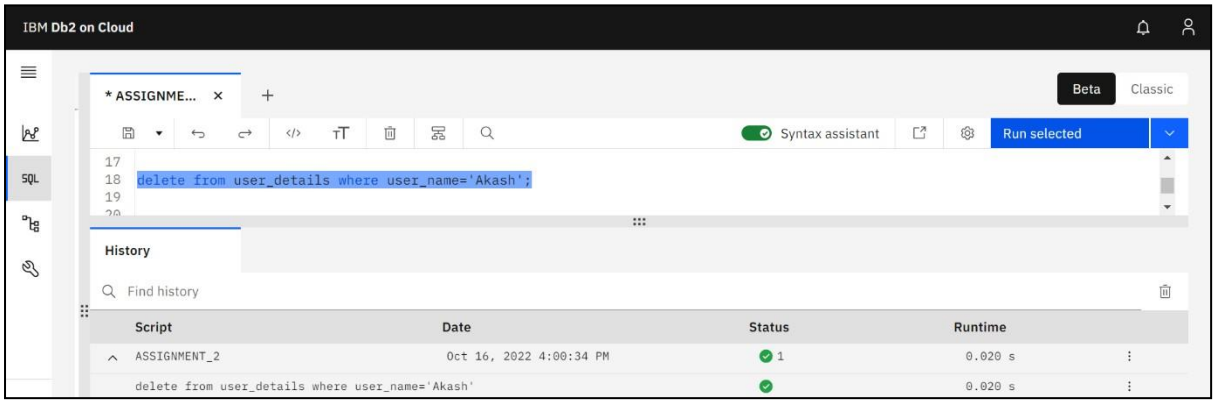
User_details table:



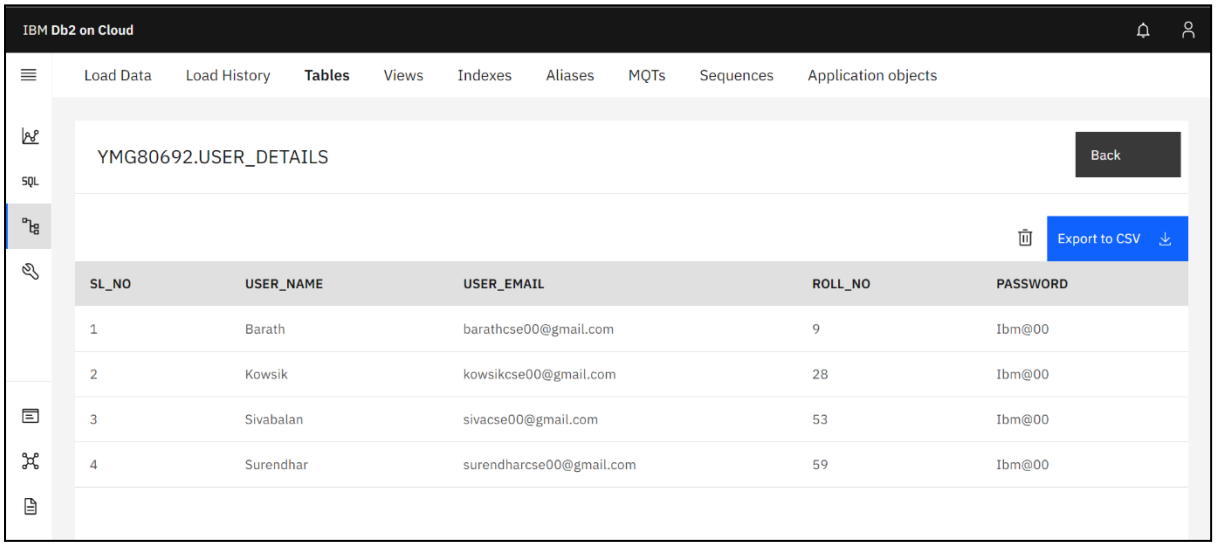
SL_NO	USER_NAME	USER_EMAIL	ROLL_NO	PASSWORD
1	Barath	barathcse00@gmail.com	9	Ibm@00
2	Kowsik	kowsikcse00@gmail.com	28	Ibm@00
3	Sivabalan	sivacse00@gmail.com	53	Ibm@00
4	Surendhar	surendharcse00@gmail.com	59	Ibm@00
5	Akash	akashece00@gmail.com	4	Ibm@00

2. Perform update & delete queries with the table





Updated table:



SL_NO	USER_NAME	USER_EMAIL	ROLL_NO	PASSWORD
1	Barath	barathcse00@gmail.com	9	Ibm@00
2	Kowsik	kowsikcse00@gmail.com	28	Ibm@00
3	Sivabalan	sivacse00@gmail.com	53	Ibm@00
4	Surendhar	surendharcse00@gmail.com	59	Ibm@00

3. **Connect python code to db2**

```
import ibm_db

conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=98538591-7217- 4024-
b0278baa776ffad1.c3n41cmd0nqnrk39u98.databases.appdomain.cloud;P
ORT=30875;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;PRO
T OCOL=TCPIP;UID=ymg80692;PWD=dS5CZPeX9CN20vpY ",",")
```

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**

App.py

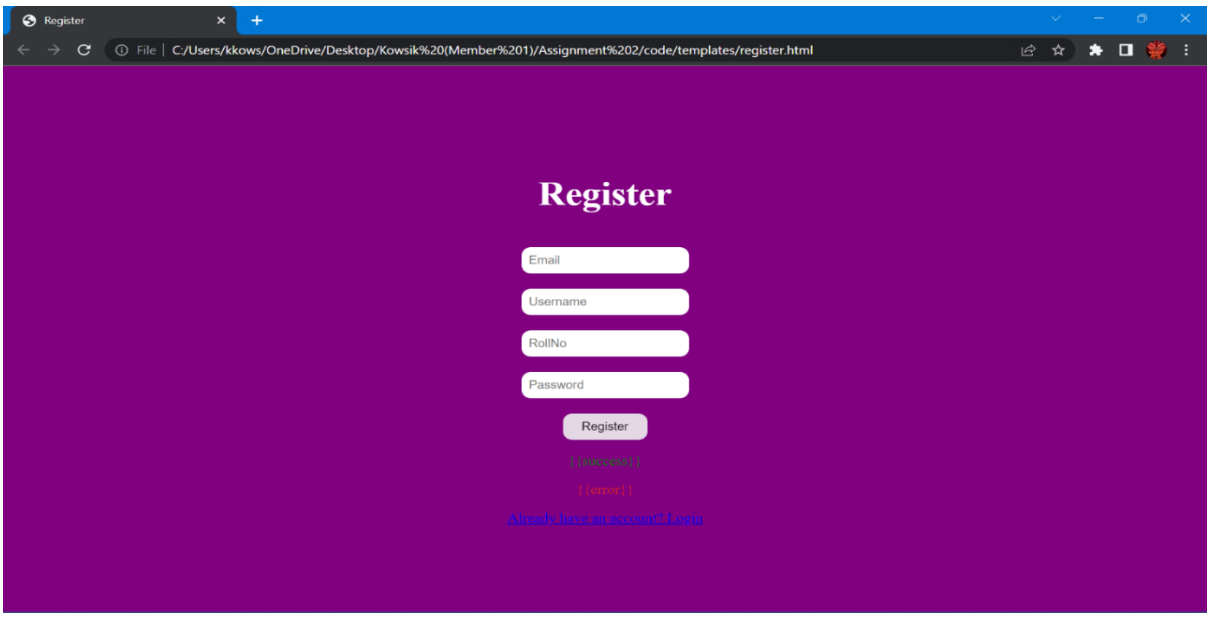
```
from flask import Flask, render_template, request, redirect, url_for, session

import ibm_db
import bcrypt
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=98538591-7217-4024-b027-
8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud;PORT=30875;SECU
RITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;PROTOCOL=TCPIP;UID=ymg8069
2;PWD=dS5CZPeX9CN20vpY ",",")
```

```
# 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')
```

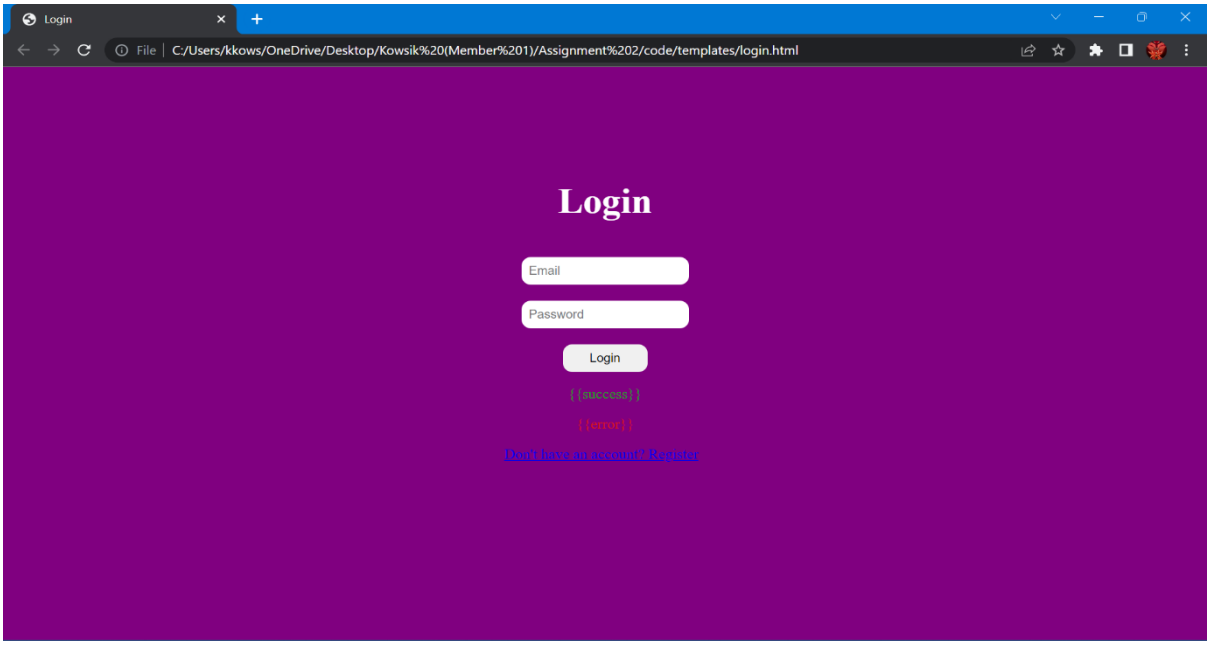
```
@app.route("/login",methods=['GET','POST'])
def login():
   if request.method == 'POST':
     email = request.form['email']
     password = request.form['password']

     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:

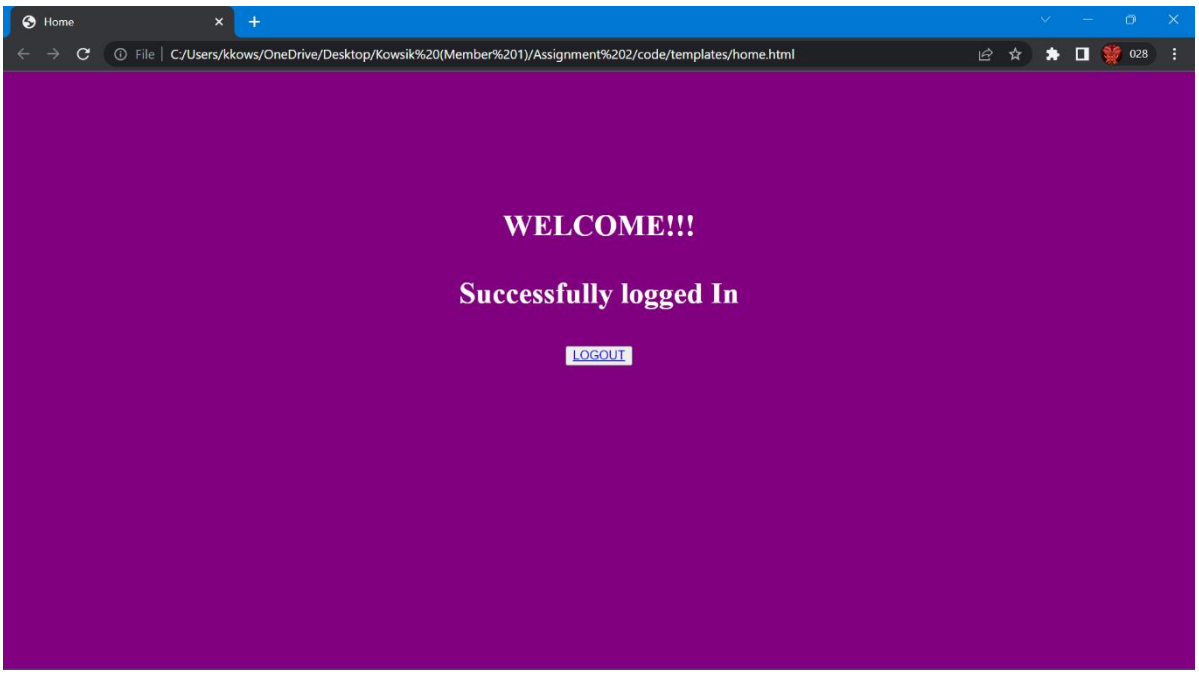
Registering:



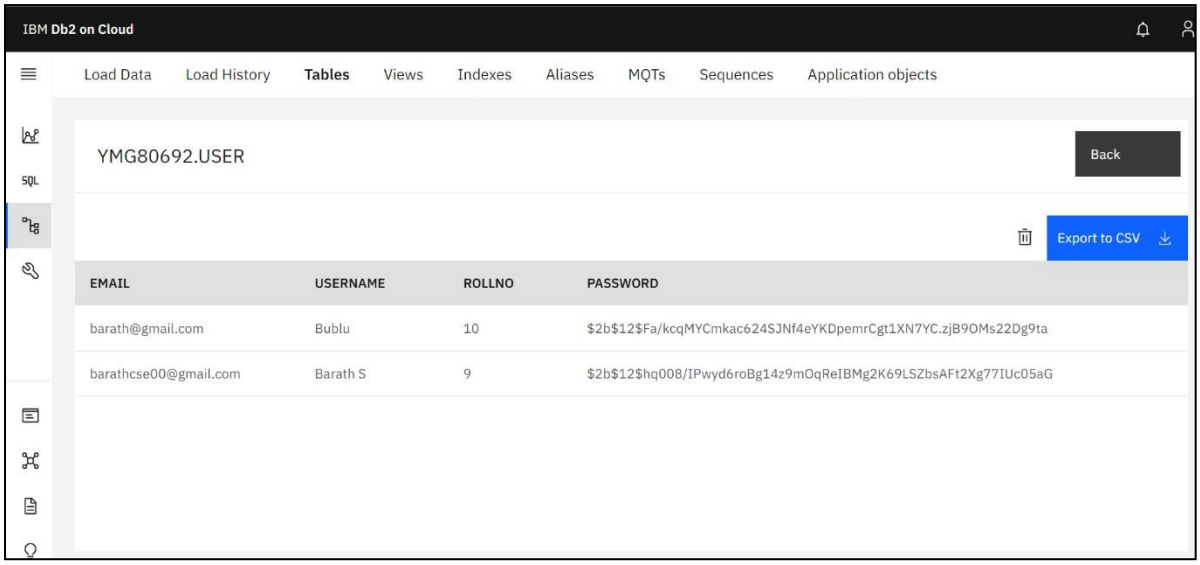
Logging in:



Home page:



Database:



EMAIL	USERNAME	ROLLNO	PASSWORD
barath@gmail.com	Bublu	10	$2b$12$Fa/kcqMYCmkac624SJNf4eYKDpemrCgt1XN7YC.zjB9OMs22Dg9ta
barathcse00@gmail.com	Barath S	9	$2b$12$hq008/IPwyd6roBg14z9mOqReIBMg2K69LSZbsAFt2Xg77IUc05aG