Akarshan S

**ASSIGNMENT-2**

|  |  |
| --- | --- |
| Team ID | PNT2022TMID05721 |
| Project Name | Project – NEWS TRACKER APPLICATION |

**SOURCE CODE**

**CRUD OPERATION IN IBM DB2:**

1. **Create Table:**

CREATE TABLE user\_table ( id INT NOT NULL PRIMARY KEY GENERATED

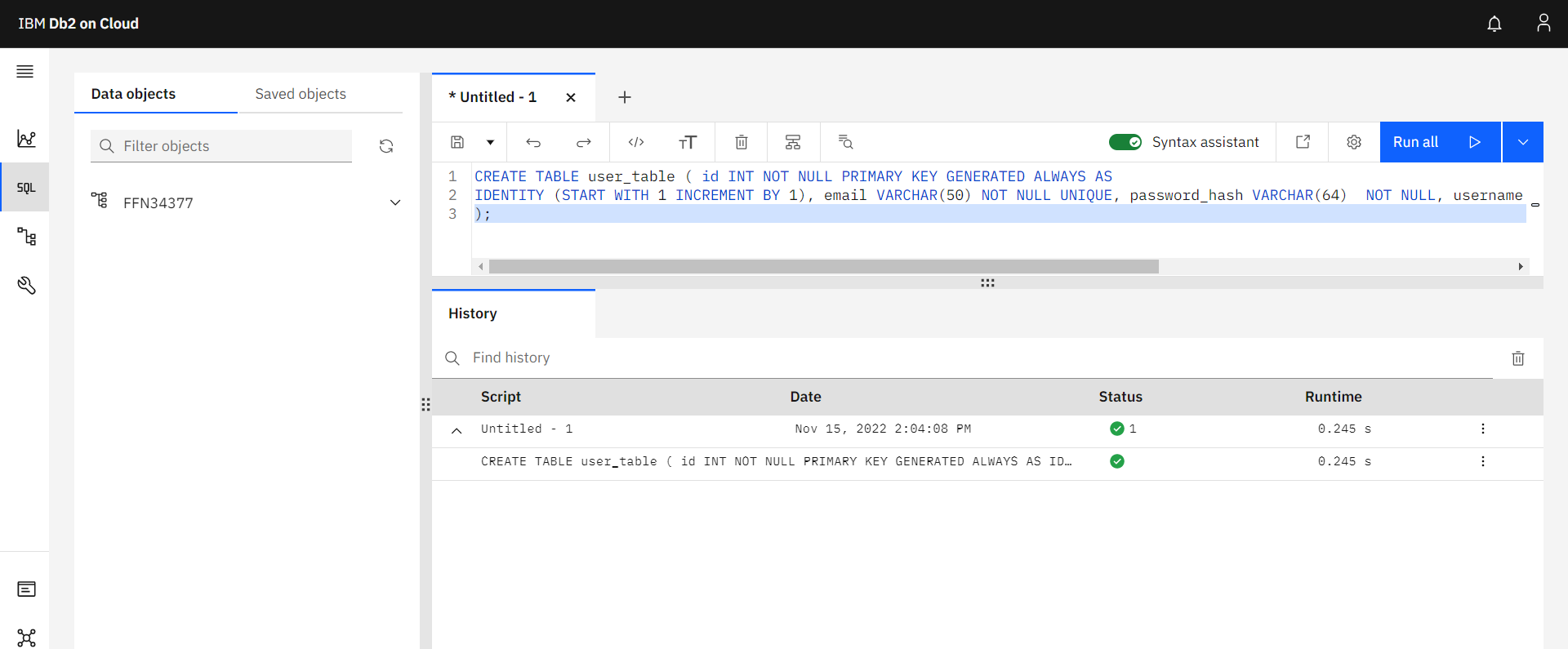
ALWAYS AS IDENTITY (START WITH 1 INCREMENT BY 1),

email VARCHAR(50) NOT NULL UNIQUE,

password\_hash VARCHAR(64) NOT NULL,

username VARCHAR(50) NOT NULL UNIQUE,

rollno VARCHAR(15) NOT NULL UNIQUE );

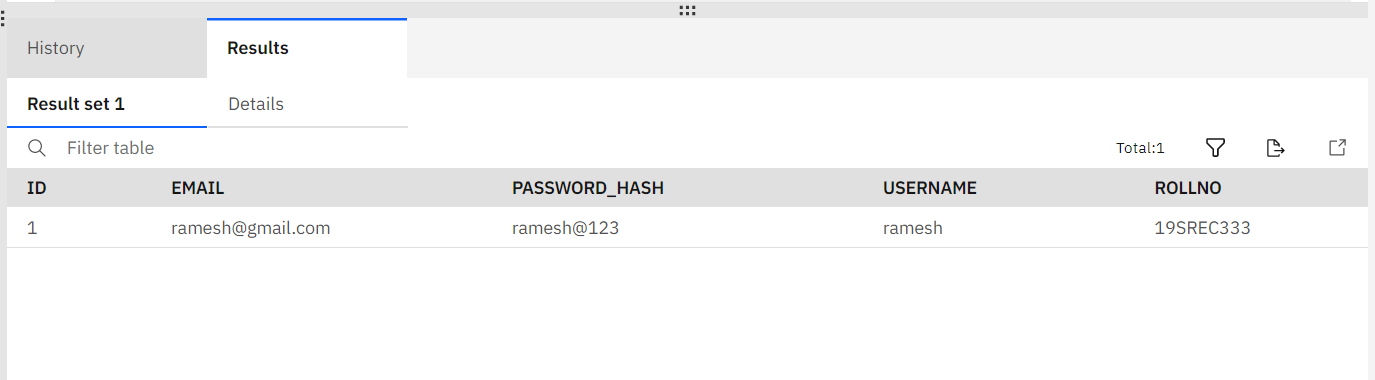


1. **Insert:**

INSERT INTO user\_table (email, password\_hash, username, rollno) VALUES ('ramesh@gmail.com', 'ramesh@123', 'ramesh', '19SREC333');

1. **Read:**

SELECT \* from user\_table;



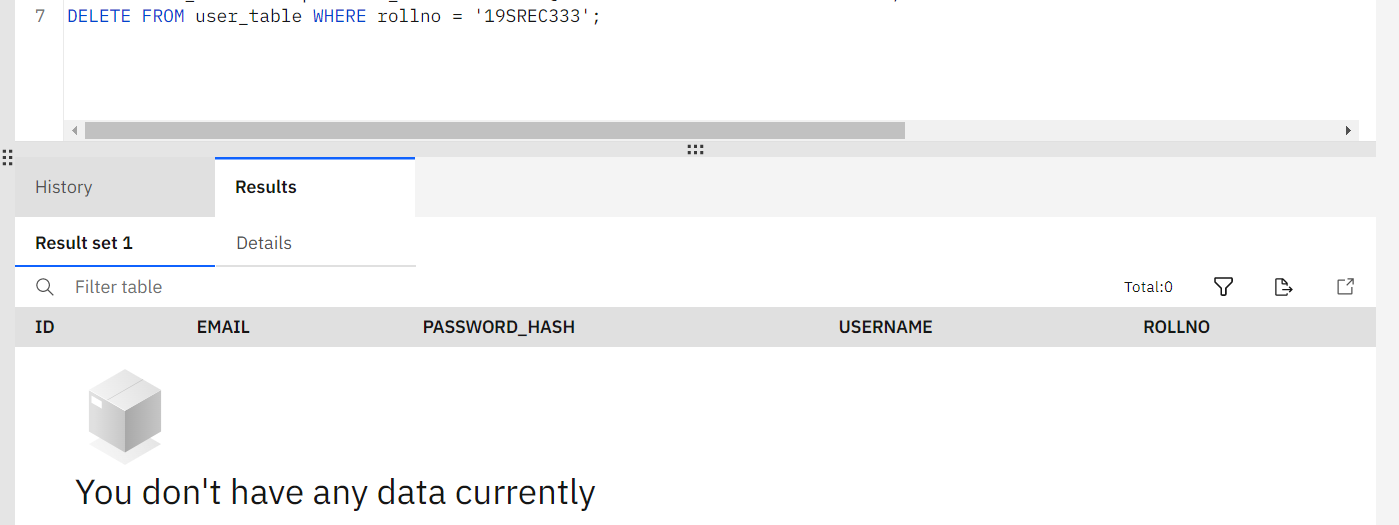
1. **Update:**

UPDATE user\_table SET password\_hash = 'ramesh@890' WHERE rollno = '19SREC333';



1. **Delete:**

DELETE FROM user\_table WHERE rollno = '19SREC333';



**FLASK APP**

<!DOCTYPE html>

<html>

<head>

<title>NewsApp</title>

<link rel="stylesheet" type="text/css" href="../static/css/login.css">

<link href="https://fonts.googleapis.com/css2?family=Jost:wght@500&display=swap" rel="stylesheet">

</head>

<body>

<div class="main">

<input type="checkbox" id="chk" aria-hidden="true">

<div class="signup">

<form action="/register" method="post">

<div class="msg">{{ msg }}</div>

<label class="signName" for="chk" aria-hidden="true">Sign up</label>

<input type="text" name="username" placeholder="Enter Your Username" id="username" required></br></br>

<input type="email" name="email" placeholder="Enter Your Email ID" id="email" required></br></br>

<input type="password" name="password" placeholder="Enter Your Password" id="password" required></br></br>

<button type="submit" id="button" > Register </button>

</form>

</div>

<div class="login">

<form action="/login" method="post">

<div class="msg">{{ msg }}</div>

<label class="loginName" for="chk" aria-hidden="true">Login</label>

<input type="text" name="username" placeholder="Enter Your Username" id="username" required></br></br>

<input type="password" name="password" placeholder="Enter Your Password" id="password" required></br></br>

<button>Login</button>

</form>

</div>

</div>

</body>

</html>

import re

import ibm\_db

from flask import Flask, redirect, render\_template, request, session, url\_for

from newsapi import NewsApiClient

# init flask app

app = Flask(\_name\_)

app.secret\_key = 'a'

conn= ibm\_db.connect("DATABASE=bludb;HOSTNAME=2f3279a5-73d1-4859-88f0-a6c3e6b4b907.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud;PORT=30756;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=nkk12877;PWD=UUGFBq21gm6EiMIz",'','')

# Init news api

newsapi = NewsApiClient(api\_key='70fdb9ba81ba40b6bda148e672898bd9')

# helper function

def get\_sources\_and\_domains():

all\_sources = newsapi.get\_sources()['sources']

sources = []

domains = []

for e in all\_sources:

id = e['id']

domain = e['url'].replace("http://", "")

domain = domain.replace("https://", "")

domain = domain.replace("www.", "")

slash = domain.find('/')

if slash != -1:

domain = domain[:slash]

sources.append(id)

domains.append(domain)

sources = ", ".join(sources)

domains = ", ".join(domains)

return sources, domains

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

def login():

global userid

msg = ''

if request.method == 'POST' :

username = request.form['username']

password = request.form['password']

sql = "SELECT \* FROM users WHERE username =? AND password=?"

stmt = ibm\_db.prepare(conn, sql)

ibm\_db.bind\_param(stmt,1,username)

ibm\_db.bind\_param(stmt,2,password)

ibm\_db.execute(stmt)

account = ibm\_db.fetch\_assoc(stmt)

print (account)

if account:

session['loggedin'] = True

session['id'] = account['USERNAME']

userid= account['USERNAME']

session['username'] = account['USERNAME']

msg = 'Logged in successfully !'

msg = 'Logged in successfully !'

return redirect(url\_for('home'))

else:

msg = 'Incorrect username / password !'

return render\_template('login-register.html', msg = msg)

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

def register():

msg = ''

if request.method == 'POST' :

username = request.form['username']

email = request.form['email']

password = request.form['password']

sql = "SELECT \* FROM users WHERE username =?"

stmt = ibm\_db.prepare(conn, sql)

ibm\_db.bind\_param(stmt,1,username)

ibm\_db.execute(stmt)

account = ibm\_db.fetch\_assoc(stmt)

print(account)

if account:

msg = 'Account already exists !'

elif not re.match(r'[^@]+@[^@]+\.[^@]+', email):

msg = 'Invalid email address !'

elif not re.match(r'[A-Za-z0-9]+', username):

msg = 'name must contain only characters and numbers !'

else:

insert\_sql = "INSERT INTO users 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, password)

ibm\_db.execute(prep\_stmt)

msg = 'You have successfully registered !'

elif request.method == 'POST':

msg = 'Please fill out the form !'

return render\_template('login-register.html', msg = msg)

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

def home():

if request.method == "POST":

sources, domains = get\_sources\_and\_domains()

keyword = request.form["keyword"]

related\_news = newsapi.get\_everything(q=keyword,

sources=sources,

domains=domains,

language='en',

sort\_by='relevancy')

no\_of\_articles = related\_news['totalResults']

if no\_of\_articles > 100:

no\_of\_articles = 100

all\_articles = newsapi.get\_everything(q=keyword,

sources=sources,

domains=domains,

language='en',

sort\_by='relevancy',

page\_size = no\_of\_articles)['articles']

return render\_template("home.html", all\_articles = all\_articles,

keyword=keyword)

else:

top\_headlines = newsapi.get\_top\_headlines(country="in", language="en")

total\_results = top\_headlines['totalResults']

if total\_results > 100:

total\_results = 100

all\_headlines = newsapi.get\_top\_headlines(country="in",

language="en",

page\_size=total\_results)['articles']

return render\_template("home.html", all\_headlines = all\_headlines)

return render\_template("home.html")

if \_name\_ == "\_main\_":

app.run(debug = True)