

## Assignment -2

### Connect with Database Assignment

|                     |                 |
|---------------------|-----------------|
| Assignment Date     | 09 October 2022 |
| Student Name        | S.Dineshwaran   |
| Student Roll Number | 621319104014    |
| Maximum Marks       | 2 Marks         |

#### Question-1:

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

The screenshot shows the IBM Db2 Cloud SQL editor interface. The top toolbar includes a 'Run all' button. The main editor area contains the following SQL script:

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

Below the editor, the 'History' tab is active, displaying a table of executed scripts:

| 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 |

The screenshot shows the 'Structure' view of the 'USER' table. The table has the following columns:

| EMAIL         | USERNAME | ROLLNO | PASSWORD |
|---------------|----------|--------|----------|
| PHB43134.USER |          |        |          |

2.Perform UPDATE, DELETE Queries with User table

The first screenshot shows the execution of an INSERT query into the 'USER' table:

```
1 INSERT INTO USER VALUES('ram@gmail.com','Ram',12345,'HEll@123');
```

The second screenshot shows the execution of another INSERT query:

```
1 INSERT INTO USER VALUES('suresh@gmail.com','Suresh',67890,'Sui@123');
```

## Table View:

| IBM Db2 on Cloud                                                                              |          |        |          |               |
|-----------------------------------------------------------------------------------------------|----------|--------|----------|---------------|
| Load Data Load History <b>Tables</b> Views Indexes Aliases MQTs Sequences Application objects |          |        |          |               |
| PHB43134.USER                                                                                 |          |        |          | Back          |
|                                                                                               |          |        |          | Export to CSV |
| EMAIL                                                                                         | USERNAME | ROLLNO | PASSWORD |               |
| ram@gmail.com                                                                                 | Ram      | 12345  | HEll@123 |               |
| suresh@gmail.com                                                                              | Suresh   | 67890  | Sur@123  |               |

## UPDATE:

\*Untitled - 1 x +

Beta Classic

☒ Syntax assistant   Run all v

1 UPDATE USER SET ROLLNO=87653 WHERE USERNAME='Ram';



## Table View:


| PHB43134.USER    |          |        |          | Back          |
|------------------|----------|--------|----------|---------------|
|                  |          |        |          | Export to CSV |
| EMAIL            | USERNAME | ROLLNO | PASSWORD |               |
| ram@gmail.com    | Ram      | 87653  | HEll@123 |               |
| suresh@gmail.com | Suresh   | 67890  | Sur@123  |               |

## DELETE:

\*Untitled - 1 x +



Beta Classic

☒ Syntax assistant   Run all v

1 INSERT INTO USER VALUES('ram@gmail.com','Ram',12345,'HEll@123');

TABLE View:

| PHB43134.USER |          |        |          | Back                                                                                                                                                                                                  |
|---------------|----------|--------|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|               |          |        |          |  <a href="#">Export to CSV</a>  |
| EMAIL         | USERNAME | ROLLNO | PASSWORD |                                                                                                                                                                                                       |
| ram@gmail.com | Ram      | 87653  | HEll@123 |                                                                                                                                                                                                       |

### 3. Connect python with db2.

Solution:

```
import ibm_db
```

```
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=32733;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;PROTOCOL=TCPIP;UID=phb43134;PWD=mTTnIE45Raj9A0rr", "", "")
```

**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=DigiCertGlobalRootCA.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(prepare_stmt, 1, username)
            ibm_db.bind_param(prepare_stmt, 2, email)
            ibm_db.bind_param(prepare_stmt, 3, hash)
            ibm_db.bind_param(prepare_stmt, 4, rollNo)
            ibm_db.execute(prepare_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:

The image displays two screenshots of a web browser showing a registration form titled "Register".

**Top Screenshot:** The form is empty. It has four input fields: "Email", "Username", "RollNo", and "Password". Below the fields is a "Register" button and a link that says "Already have an account? Login".

**Bottom Screenshot:** The form is filled with the following values: "Email" is "dineshwaran@gmail.com", "Username" is "Dineshwaran", "RollNo" is "12345", and "Password" is masked with "\*\*\*\*\*". The "Register" button and the "Already have an account? Login" link are still visible below the fields.

Service Details - IBM Cloud x IBM Db2 on Cloud x IBM Db2 on Cloud x Register x +

127.0.0.1:5000/register

Gmail YouTube Google Font Awesome Icons IBM ANNA Univers... color picker - Goog... WhatsApp Other favorites

## Register

Email

Username

RollNo

Password

Register

You can login

[Already have an account? Login](#)

## Login

Login

[Don't have an account? Register](#)

Service: x IBM D: x Docu: x how: x SQL C: x SQL S: x sql co: x sql co: x Online: x Insert: x SQL: x alter: x SQL S: x +

127.0.0.1:5000

Gmail YouTube Google Font Awesome Icons IBM ANNA Univers... color picker - Goog... WhatsApp Other favorites

## WELCOME You are Logged In

LOGOUT

### Database:

| PHB43134.USER         |             |        |                                                                | Back          |
|-----------------------|-------------|--------|----------------------------------------------------------------|---------------|
|                       |             |        |                                                                | Export to CSV |
| EMAIL                 | USERNAME    | ROLLNO | PASSWORD                                                       |               |
| dineshwaran@gmail.com | Dineshwaran | 12345  | \$2b\$12\$dplal55ry6J94nrPvZRfODt9r45.dFEne9aJo35YQg/gQ48PJLKi |               |