

## ASSINGMENT -2

Assingment date	19 SEPTEMBER 2022
Student name	Dheekshana S
Student roll no	510419104023

- 1.Create user table with user with email,username,password,roll no
2. perform update ,delete queries with user table
- 3.connect python code to database2
- 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 amnd password.if the user is valid show the welcome page

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

```
CREATE TABLE user (  
    roll_number int,  
    username varchar(300),  
    email varchar(300),  
    password varchar(300)  
);
```



roll_number	username	email	password
empty			

Show all X

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

### INSERT Statement:

INSERT INTO user

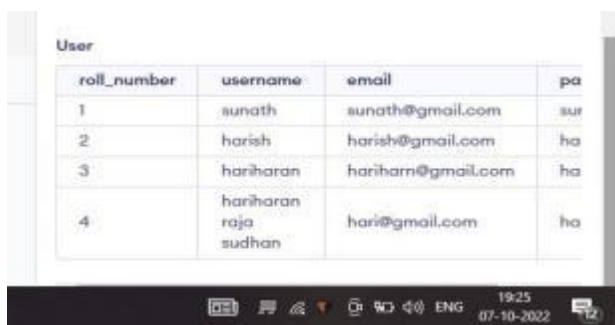
( roll\_number, username ,email, password) VALUES

(1, 'sunanth', ' sunanth@gmail.com','sunanth15'),

(2, 'harish', ' harish@gmail.com','harish123'),

(3, 'hariharan', 'hariharn@gmail.com', hari123'),

(4, 'hariharan raja sudhan', 'hari@gmail.com', hari0123');



The screenshot shows a mobile application interface with a table titled 'User'. The table has four columns: 'roll\_number', 'username', 'email', and 'pa'. It contains four rows of data.

roll_number	username	email	pa
1	sunath	sunath@gmail.com	sur
2	harish	harish@gmail.com	ha
3	hariharan	hariharn@gmail.com	ha
4	hariharan raja sudhan	hari@gmail.com	ha

### UPDATE Statement:

UPDATE users

SET username = 'sunanth'

WHERE roll\_number = '4'



The screenshot shows the same mobile application interface with the 'User' table. The data in the table has been updated: the username for roll\_number 4 is now 'sunanth' and the email is 'hari@gmail.com'.

roll_number	username	email	pa
1	sunath	sunath@gmail.com	sur
2	harish	harish@gmail.com	ha
3	hariharan	hariharn@gmail.com	ha
4	sunanth	hari@gmail.com	ha

## DELETE Statemnet:

```
insert into user values(4,'aa','aaa@gmail.com','aasdfg2') ;
```

```
delete from user where roll_number='4'
```

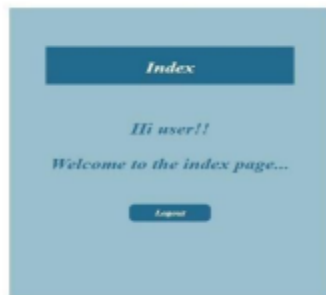


The screenshot shows a web application interface with a table titled 'User'. The table has four columns: 'roll\_number', 'username', 'email', and 'password'. There are three rows of data. The bottom of the screenshot shows a Windows taskbar with the date 07-10-2022 and time 19:29.

roll_number	username	email	password
1	sunath	sunath@gmail.com	sun
2	harish	harish@gmail.com	ha
3	hariharan	hariharan@gmail.com	ha

## 3.Connect python with db2

```
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=824dfd4d-99de-440d-9991-629c01b3832d.bs2io90108kqb1od8lcg.databases.appdomain.cloud;PORT=30119;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=lvq43963;PWD=BsnsG112sBgIRhVN",'','')
```



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

```
from flask import Flask, render_template, request, redirect, url_for, session  
from flask_mysqldb import MySQL import MySQLdb.cursors
```

```

import reapp = Flask(__name__)      app.secret_key
= 'your secret key'  app.config['MYSQL_HOST'] =
'localhost'  app.config['MYSQL_USER'] = 'root'
app.config['MYSQL_PASSWORD'] = 'your password'
app.config['MYSQL_DB'] = 'geeklogin'      mysql =
MySQL(app)

@app.route('/')

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

def login():

    msg = "if request.method == 'POST' and 'username' in request.form and
'password' in request.form:

        username = request.form['username']

password = request.form['password']

        cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)

        cursor.execute('SELECT * FROM accounts WHERE username = % s
AND password = % s', (username, password,
))
        account = cursor.fetchone()

if account:

        session['loggedin'] = True

session['id'] = account['id']
session['username'] = account['username']

msg = 'Logged in successfully !'

        return render_template('index.html', msg = msg)

    else:

        msg = 'Incorrect username / password !'

```

```
return render_template('login.html', msg = msg)
```

```
@app.route('/logout') def
```

```
logout():
```

```
    session.pop('logged_in', None)
```

```
    session.pop('id', None)
```

```
    session.pop('username', None)    return
```

```
    redirect(url_for('login'))
```

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

```
register():
```

```
    msg = "
```

```
    if request.method == 'POST' and 'username' in request.form and 'password'
    in request.form and 'email' in request.form :
```

```
        username = request.form['username']
```

```
        password = request.form['password']        email =
```

```
        request.form['email']
```

```
        cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
```

```
        cursor.execute('SELECT * FROM accounts WHERE username = % s',
        (username, ))
```

```
        account = cursor.fetchone()
```

```
        if account:
```

```
            msg = 'Account already exists !'
```

```
        elif not re.match(r'^[a-zA-Z0-9]+@[a-zA-Z0-9]+\.[a-zA-Z]+', email):
```

```

msg = 'Invalid email address !' elif not re.match(r'[A-Za-z0-
9]'+, username):
    msg = 'Username must contain only characters
and numbers !' elif not username or not password or not email:
    msg = 'Please fill out the form !'

else:
    cursor.execute('INSERT INTO accounts VALUES (NULL, %
s, % s, % s)', (username, password, email, ))
    mysql.connection.commit()
    msg = 'You have successfully registered !'

elif request.method == 'POST':
    msg = 'Please fill out the form !' return
render_template('register.html', msg = msg)

```

