

## Assignment -2

Assignment Date	19 September 2022
Student Name	VINOTHKUMAR S
Student Roll Number	310119104088
Maximum Marks	2 Marks

### Question-1:

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

```
CREATE TABLE user3 ( roll_number  
    int,  
    username varchar(300), email  
    varchar(300),  
    password varchar(300)  
);
```

The screenshot shows a web-based SQL editor. At the top, there's a tab labeled 'Input'. Below it, a toolbar contains icons for undo, redo, and a 'Run SQL' button. The main text area contains the following SQL code:

```
-- Online SQL Editor to Run SQL Online.  
-- Use the editor to create new tables, insert data and all other SQL operations.  
  
CREATE TABLE users(roll_number int, username varchar(300), email varchar(300),  
password varchar(300));
```

1. Perform UPDATE, DELETE Queries with user table

### INSERT Statement:

```
INSERT INTO users3( roll_number, username ,email, password) VALUES  
(1, 'Vinothkumar', 'vinothkuma415a@gmail.com','vinoth88'),  
(2, 'Sasirajan', 'sasirajan@gmail.com','sasi71'),  
(3, 'Sutharshan', 'sutharshan@gmail.com','sutharshan81'), (4,  
'Yuvaraj', 'yuvi@gmail.com','yuvaraj94');
```

Input

Run SQL

```
CREATE TABLE users3(roll_number int, username varchar(300), email varchar(300), password varchar(300));

INSERT INTO users3( roll_number, username ,email, password) VALUES (1, 'Vinothkumar', 'vinothkuma415a@gmail.com','vinoth88'), (2, 'Sasirajan', 'sasirajan@gmail.com','sasi71'), (3, 'Sutharshan', 'sutharshan@gmail.com','sutharshan81'), (4, 'Yuvaraj', 'yuvi@gmail.com','yuvaraj94');
```

Output

SQL query successfully executed. However, the result set is empty.

Users3

roll_number	username	email
1	Vinothkumar	vinothkuma415a@g
2	Sasirajan	sasirajan@gmail.co
3	Sutharshan	sutharshan@gmail.c
4	Yuvaraj	yuvi@gmail.com



### UPDATE Statement:

UPDATE users4 SET username = 'Sutharshan' WHERE roll\_number = '2';

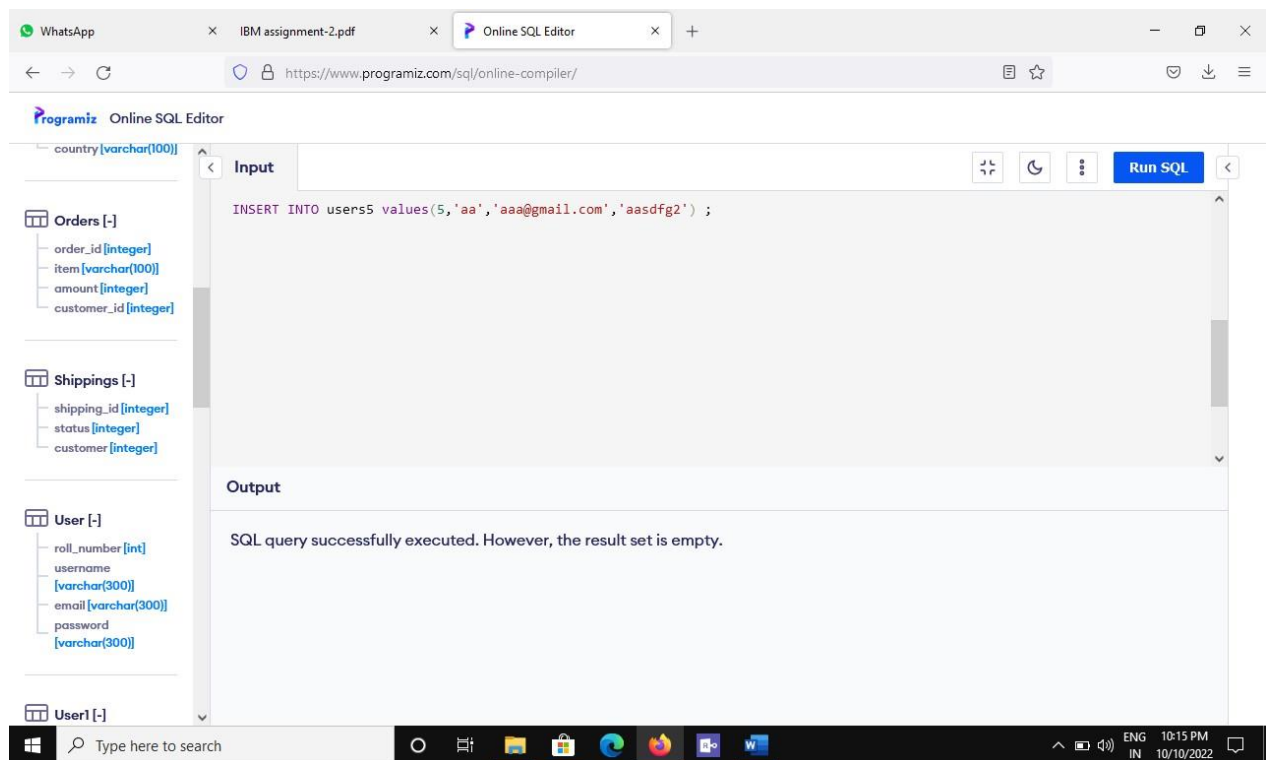
The screenshot shows a web browser with the 'Online SQL Editor' tab active. The URL is <https://www.programiz.com/sql/online-compiler/>. The interface includes a sidebar with database schema information for 'country', 'Orders', 'Shippings', 'User', and 'User1'. The main area has an 'Input' section with the SQL query: `UPDATE users4 SET username = 'Sutharshan' WHERE roll_number = '2';` and a 'Run SQL' button. The 'Output' section displays the message: 'SQL query successfully executed. However, the result set is empty.'

### Users4

roll_number	username	email
1	Vinothkumar	vinothkuma415a@g
2	Sutharshan	sasirajan@gmail.co
3	Sutharshan	sutharshan@gmail.x
4	Yuvaraj	yuvi@gmail.com

### Insert Statement:

`INSERT INTO users5 values(5,'aa','aaa@gmail.com','aasdfg2') ;`

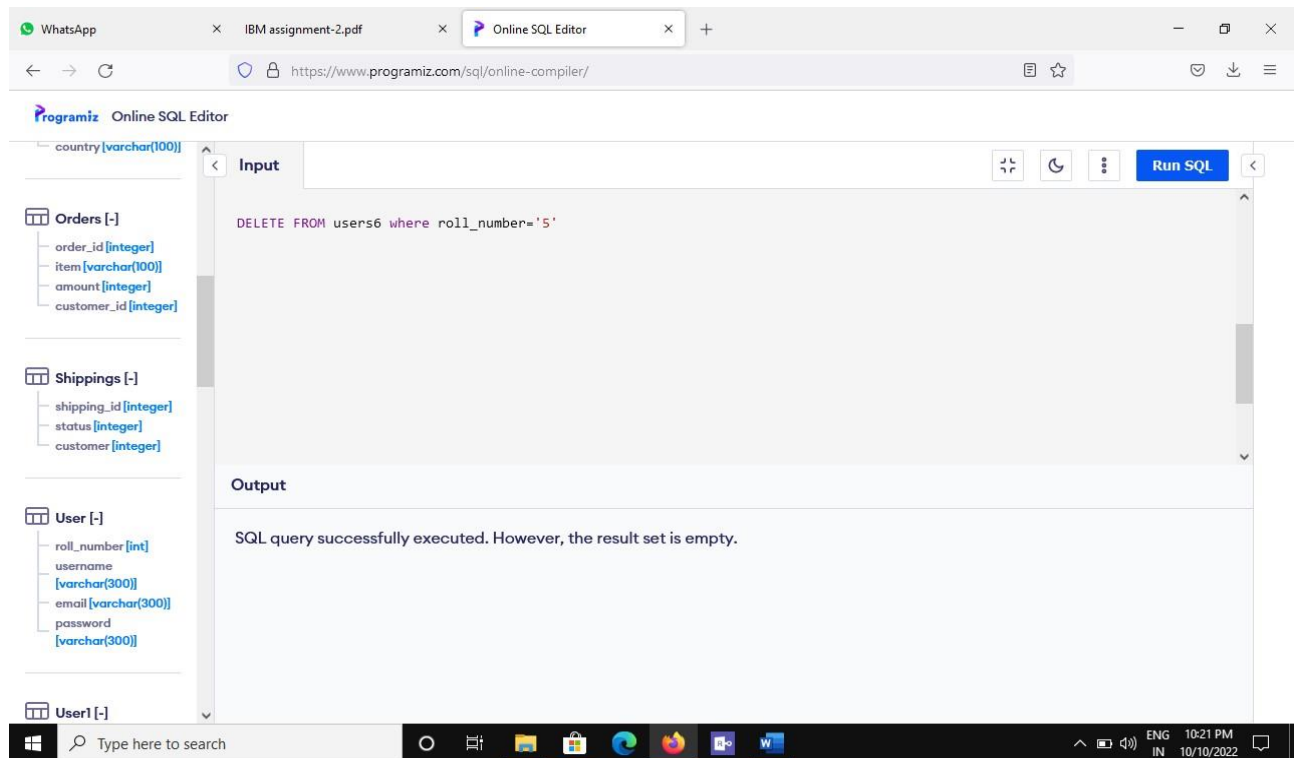


**Users5**

roll_number	username	email
1	Vinothkumar	vinothkuma415a@gi
2	Sutharshan	sasirajan@gmail.co
3	Sutharshan	sutharshan@gmail.x
4	Yuvaraj	yuvi@gmail.com
5	aa	aaa@gmail.com

### DELETE Statement:

DELETE FROM users6 where roll\_number='5'



Users6		
roll_number	username	email
1	Vinothkumar	vinothkuma415a@g
2	Sutharshan	sasirajan@gmail.co
3	Sutharshan	sutharshan@gmail.c
4	Yuvaraj	yuvi@gmail.com

Connect python with db2

```
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=824dfd4d-99de-440d-
9991-
629c01b3832d.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=30119;SE
CURITY=SSL
;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=lvq43963;PWD=BsnsG112s
BgIRhVN",' ');")
```

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

```

from flask_mysql 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('loggedin', 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'[^@]+@[^@]+\.[^@]+', 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()

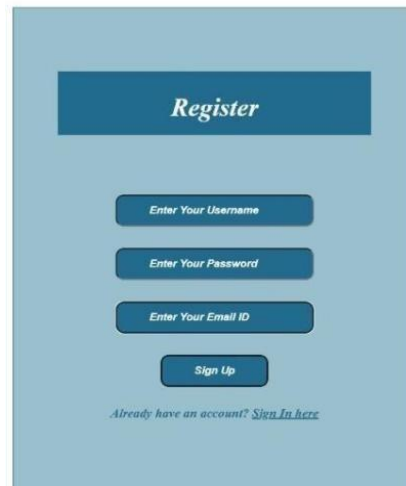
        msg = 'You have successfully registered !'

    elif request.method == 'POST':

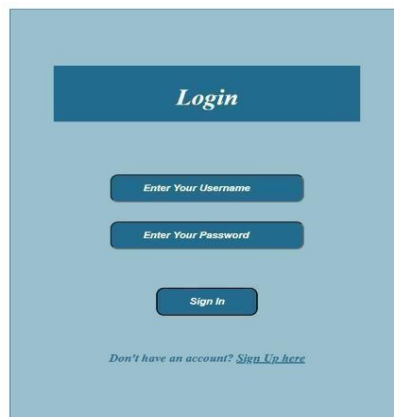
```

```
msg = 'Please fill out the form !'
```

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



A registration form with a light blue background. At the top, a dark blue header contains the word "Register" in white. Below the header, there are three input fields with dark blue borders and light blue backgrounds, each with a dark blue label: "Enter Your Username", "Enter Your Password", and "Enter Your Email ID". Below these fields is a dark blue "Sign Up" button. At the bottom, there is a link that says "Already have an account? [Sign In here](#)".



A login form with a light blue background. At the top, a dark blue header contains the word "Login" in white. Below the header, there are two input fields with dark blue borders and light blue backgrounds, each with a dark blue label: "Enter Your Username" and "Enter Your Password". Below these fields is a dark blue "Sign In" button. At the bottom, there is a link that says "Don't have an account? [Sign Up here](#)".



An index page with a light blue background. At the top, a dark blue header contains the word "Index" in white. Below the header, there is a dark blue "Hi user!!" message. Below that is a dark blue "Welcome to the index page..." message. At the bottom, there is a dark blue "Logout" button.

3) Write a flask program which should display resume details and also have upload resume option by using file uploading.



**UPLOAD.PY:**

```
from flask import *
app = Flask(__name__)

@app.route('/') def
upload():
    return render_template("file_upload_form.html")

@app.route('/success', methods = ['POST']) def success():    if
request.method == 'POST':                f = request.files['file']
f.save(f.filename)        return render_template("success.html",
name = f.filename)

if __name__ == '__main__':
app.run(debug = True)
```

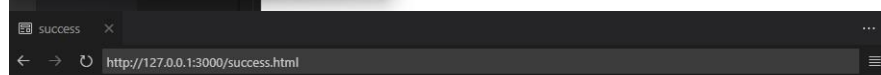
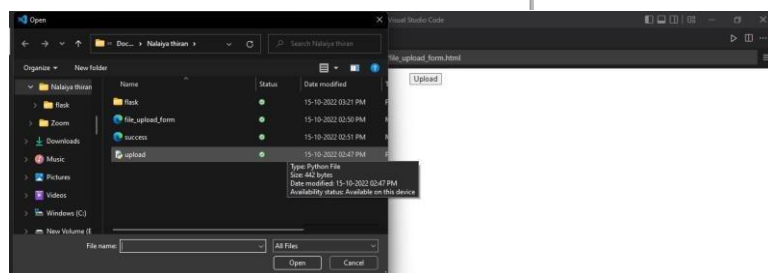
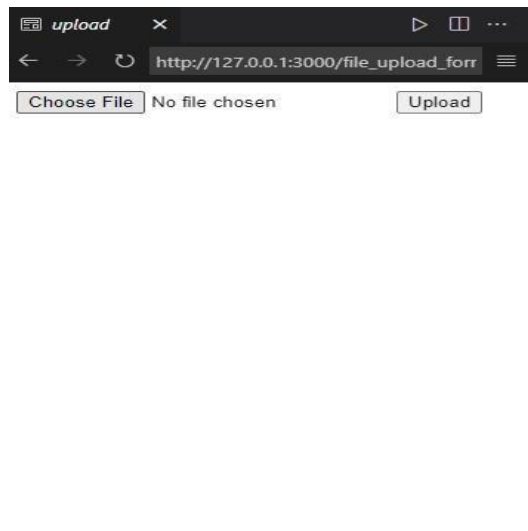
**FILE\_UPLOAD\_FORM.HTML:**

```
<html>
<head>
    <title>upload</title>
</head>
<body>
    <form action = "/success" method = "post" enctype="multipart/form-data">
<input type="file" name="file" />
    <input type = "submit" value="Upload">
    </form>
</body>
</html>
```

**SUCCESS.HTML:**

```
<html>
<head>
<title>success</title>
</head>
<body>
<p>File uploaded successfully</p>
<p>File Name: {{ name }}</p>
</body>
</html>
```

**OUTPUT:**



File Name: {{file\_upload\_form}}