

Assignment Date	12 September 2022
Student Name	Sakthivel G
Student Roll Number	422619106015
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)
);
```

>

Input

Run SQL

<

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

Output

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

1. Perform UPDATE, DELETE Queries with user table

INSERT Statement:

```
INSERT INTO users3( roll_number, username ,email, password) VALUES
(1, 'Nivetha', 'nivetha80@gmail.com','nivetha80'),
(2, 'madhu', 'madhu@gmail.com','madhu1'),
(3, 'akash', 'akash02@gmail.com','akash02'),
(4, 'pugal80', 'pugal06@gmail.com','pugal02'),
);
```

Input

```
INSERT INTO user3( roll_number, username ,email, password) VALUES (1, 'Nivetha', 'nivetha80@gmail.com','nivetha80'),
(2, 'madhu', 'madhu@gmail.com','madhu1'),
(3, 'akash', 'akash02@gmail.com','akash02'),
(4, 'pugal80', 'pugal06@gmail.com','pugal02');
```

Run SQL

Available Tables

5	Mousepad	250	2
---	----------	-----	---

Shippings

shipping_id	status	customer
1	Pending	2
2	Pending	4
3	Delivered	3
4	Pending	5
5	Delivered	1

User3

roll_number	username	email	password
1	Nivetha	nivetha80@gmail.com	nivetha80
2	madhu	madhu@gmail.com	madhu1
3	akash	akash02@gmail.com	akash02
4	pugal80	pugal06@gmail.com	pugal02

Output

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

UPDATE Statement:

UPDATE user3 SET username = 'Nivetha' WHERE roll_number = '2';

The screenshot displays a SQL query execution environment. On the left, a sidebar lists available tables: Customers, Orders, Shippings, and User3, each with its schema details. The main area is divided into 'Input' and 'Output' sections. The 'Input' section contains the SQL query: `UPDATE user3 SET username = 'Nivetha' WHERE roll_number = '2';`. The 'Output' section shows the message: 'SQL query successfully executed. However, the result set is empty.' On the right, the 'Available Tables' section displays data for three tables: a table with 4 columns (values 3, 4, 5, 5), 'Shippings' (3 columns), and 'User3' (4 columns).

Customers [-]

- customer_id [int]
- first_name [varchar(100)]
- last_name [varchar(100)]
- age [int]
- country [varchar(100)]

Orders [-]

- order_id [integer]
- item [varchar(100)]
- amount [integer]
- customer_id [integer]

Shippings [-]

- shipping_id [integer]
- status [integer]
- customer [integer]

User3 [-]

- roll_number [int]
- username

Input

```
UPDATE user3 SET username = 'Nivetha' WHERE roll_number = '2';
```

Run SQL

Output

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

Available Tables

3	Monitor	12000	3
4	Keyboard	400	1
5	Mousepad	250	2

Shippings

shipping_id	status	customer
1	Pending	2
2	Pending	4
3	Delivered	3
4	Pending	5
5	Delivered	1

User3

roll_number	username	email	password
1	Nivetha	nivetha80@gmail.com	nivetha80
2	Nivetha	madhu@gmail.com	madhu1
3	akash	akash02@gmail.com	akash02
4	pugal80	pugal06@gmail.com	pugal02

Insert Statement:

INSERT INTO user3 values(5,'aa','aaa@gmail.com','aasdfg2') ;

The screenshot displays a SQL IDE interface with the following components:

- Left Panel (Database Schema):**
 - Customers [-]**: customer_id [int], first_name [varchar(100)], last_name [varchar(100)], age [int], country [varchar(100)].
 - Orders [-]**: order_id [integer], item [varchar(100)], amount [integer], customer_id [integer].
 - Shippings [-]**: shipping_id [integer], status [integer], customer [integer].
 - User3 [-]**: roll_number [int], username.
- Input Area:** Contains the SQL statement: `INSERT INTO user3 values(5,'aa','aaa@gmail.com','aasdfg2') ;`
- Run SQL Button:** A blue button to execute the query.
- Output Area:** Displays the message: "SQL query successfully executed. However, the result set is empty."
- Available Tables:**
 - Table 1:**

4	Keyboard	400	1
5	Mousepad	250	2
 - Shippings:**

shipping_id	status	customer
1	Pending	2
2	Pending	4
3	Delivered	3
4	Pending	5
5	Delivered	1
 - User3:**

roll_number	username	email	password
1	Nivetha	nivetha80@gmail.com	nivetha80
2	Nivetha	madhu@gmail.com	madhu1
3	akash	akash02@gmail.com	akash02
4	pugal80	pugal06@gmail.com	pugal02
5	aa	aaa@gmail.com	aasdfg2

DELETE Statement:

DELETE FROM user3 where roll_number='5'

Customers [-]

customer_id [int]

first_name [varchar(100)]

last_name [varchar(100)]

age [int]

country [varchar(100)]

Orders [-]

order_id [integer]

item [varchar(100)]

amount [integer]

customer_id [integer]

Shippings [-]

shipping_id [integer]

status [integer]

customer [integer]

User3 [-]

roll_number [int]

username

Input

DELETE FROM user3 where roll_number='3'

Run SQL

Output

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

Available Tables

2	Mouse	300	4
3	Monitor	12000	3
4	Keyboard	400	1
5	Mousepad	250	2

Shippings

shipping_id	status	customer
1	Pending	2
2	Pending	4
3	Delivered	3
4	Pending	5
5	Delivered	1

User3

roll_number	username	email	password
1	Nivetha	nivetha80@gmail.com	nivetha80
2	Nivetha	madhu@gmail.com	madhu1
4	pugal80	pugal06@gmail.com	pugal02

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=BsnsG1l2s
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)

```


Register

Enter Your Username

Enter Your Password

Enter Your Email ID

Sign Up

Already have an account? [Sign In here](#)

Login

Enter Your Username

Enter Your Password

Sign In

Don't have an account? [Sign Up here](#)

Index

Hi user!!

Welcome to the index page...

Logout

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)
```

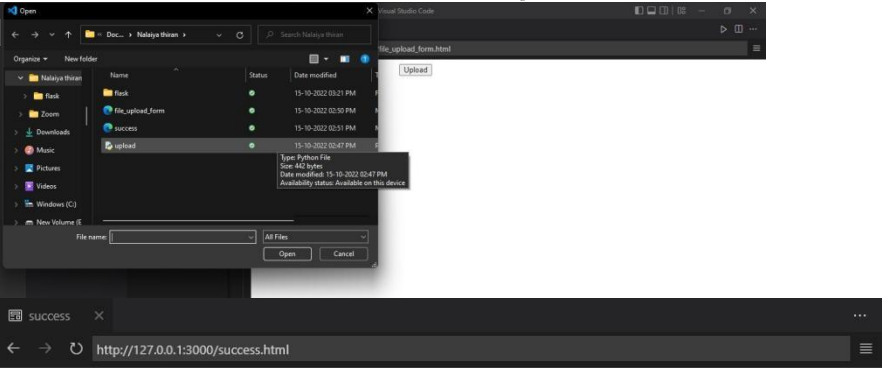
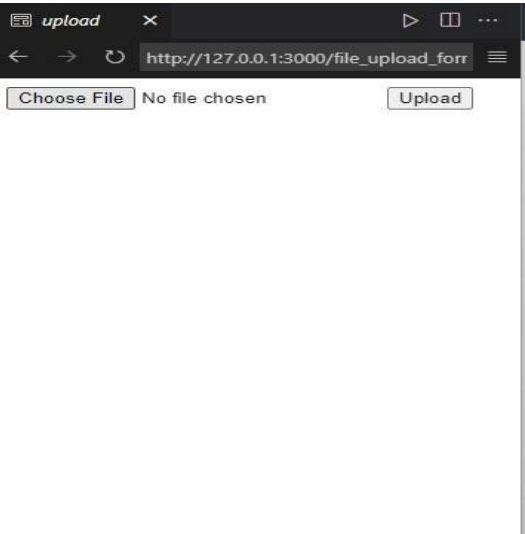
FILE_UPLOAD_FORM.HTML:

```
<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 uploaded successfully
File Name: {{file_upload_form}}