


Assignment -2

Assignment Date	19 September 2022
Student Name	Sasirajan.M
Student Roll Number	310119104071
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 an online SQL editor interface. At the top, there is a tab labeled "Input". To the right of the tab are several icons: a full-screen icon, a refresh icon, a settings icon, and a blue "Run SQL" button. Below the tab, there is a large text area containing 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

```
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@gi
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 window with the URL <https://www.programiz.com/sql/online-compiler/>. The page title is "Programiz Online SQL Editor". On the left sidebar, there is a database schema tree with the following tables and columns:

- country** [varchar(100)]
- Orders** [-]
 - order_id [integer]
 - item [varchar(100)]
 - amount [integer]
 - customer_id [integer]
- Shippings** [-]
 - shipping_id [integer]
 - status [integer]
 - customer [integer]
- User** [-]
 - roll_number [int]
 - username [varchar(300)]
 - email [varchar(300)]
 - password [varchar(300)]
- User1** [-]

The main editor area has a tab labeled "Input" containing the SQL query: `UPDATE users4 SET username = 'Sutharshan' WHERE roll_number = '2';`. A "Run SQL" button is located in the top right corner of the editor. Below the input area is an "Output" section that displays the message: "SQL query successfully executed. However, the result set is empty."

Users4

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

Insert Statement:

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

The screenshot displays a web browser window with the URL <https://www.programiz.com/sql/online-compiler/>. The browser tabs include WhatsApp, IBM assignment-2.pdf, and Online SQL Editor. The Online SQL Editor interface features a sidebar on the left with a database schema tree containing tables: country [varchar(100)], Orders [-] (with fields order_id [integer], item [varchar(100)], amount [integer], customer_id [integer]), Shippings [-] (with fields shipping_id [integer], status [integer], customer [integer]), User [-] (with fields roll_number [int], username [varchar(300)], email [varchar(300)], password [varchar(300)]), and User1 [-]. The main area is divided into 'Input' and 'Output' sections. The 'Input' section contains the SQL statement: `INSERT INTO users5 values(5,'aa','aaa@gmail.com','aasdfg2') ;`. A 'Run SQL' button is located in the top right of the input area. The 'Output' section displays the message: 'SQL query successfully executed. However, the result set is empty.' The Windows taskbar at the bottom shows the search bar, task view, and several open applications, with the system clock indicating 10:15 PM on 10/10/2022.

Users5

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

DELETE Statement:

DELETE FROM users6 where roll_number='5'

The screenshot shows a web browser window with the 'Online SQL Editor' tab active. The URL is <https://www.programiz.com/sql/online-compiler/>. The editor interface includes a sidebar with a database schema for 'country' containing tables: Orders, Shippings, User, and User1. The 'User' table schema is visible: roll_number [int], username [varchar(300)], email [varchar(300)], password [varchar(300)]. The main editor area has an 'Input' section with the SQL query: `DELETE FROM users6 where roll_number='5'`. Below the input is an 'Output' section displaying the message: 'SQL query successfully executed. However, the result set is empty.' The browser's taskbar at the bottom shows the Windows Start button, a search bar, and several application icons. The system tray on the right indicates the language is 'ENG IN', the time is '10:21 PM', and the date is '10/10/2022'.



roll_number	username	email
1	Vinothkumar	vinothkuma415a@gi
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;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=lvq43963;PWD=BsnsG112sBgIRhVN",'','')
```

```
from flask import Flask, render_template, request, redirect, url_for, session
```

```
from flask_mysqlldb 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)
```

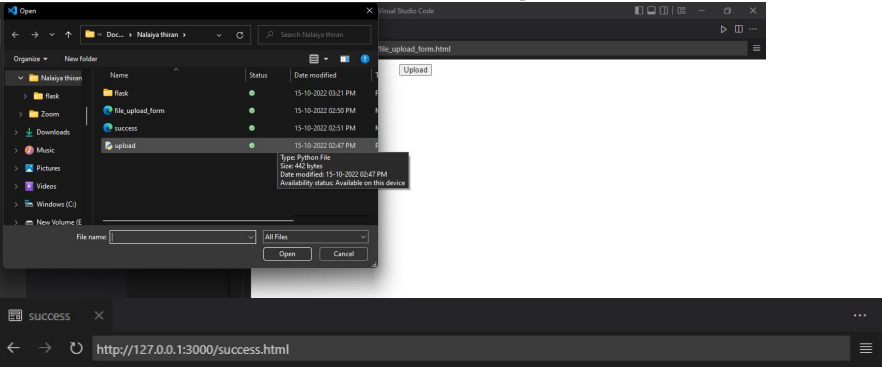
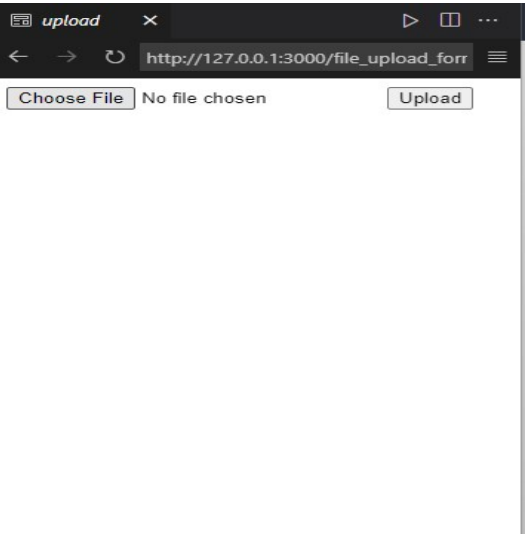
FILE_UPLOAD_FORM.HTML:

```
<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}}