

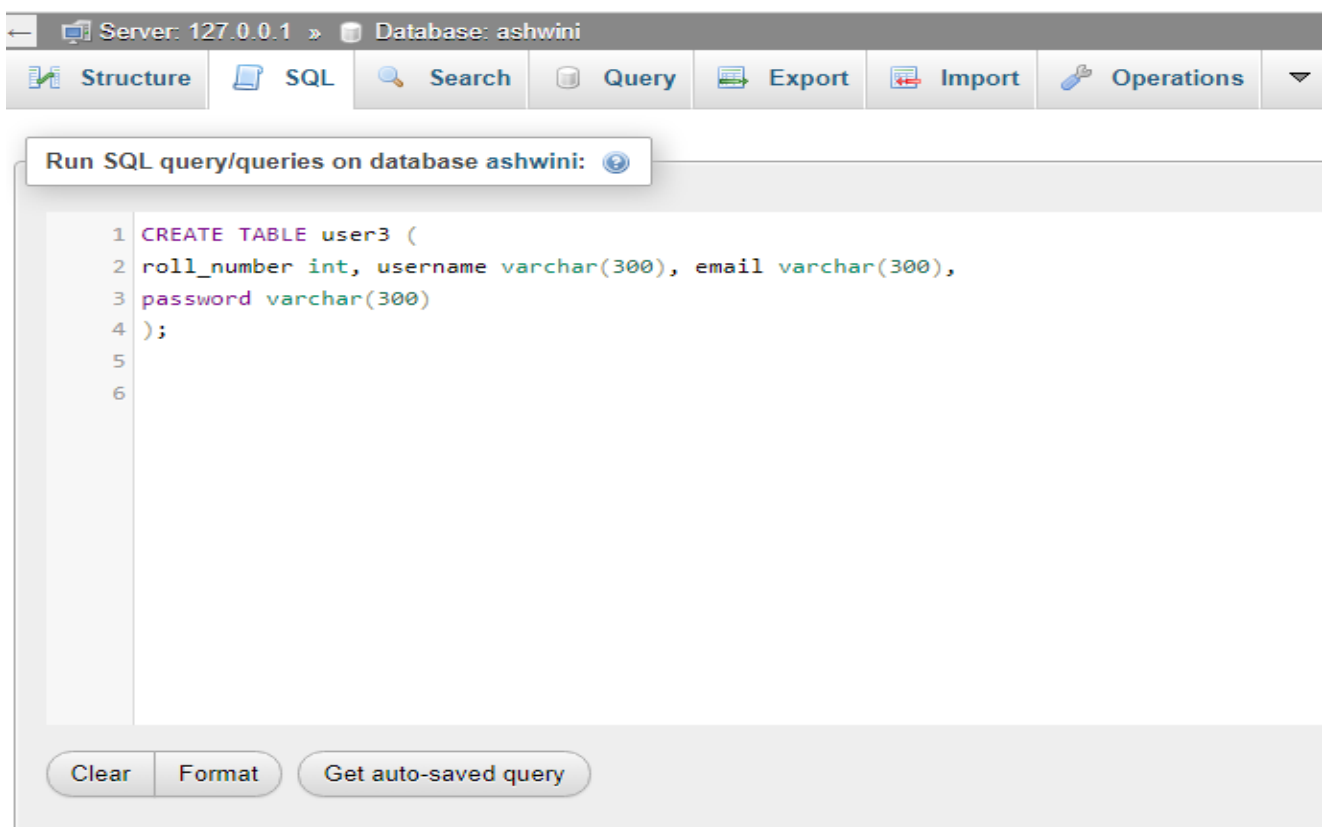
Assignment -2

Student Name	Ashwini.K
Student Roll Number	310119104013
Maximum Marks	2 Marks

Question-1:

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

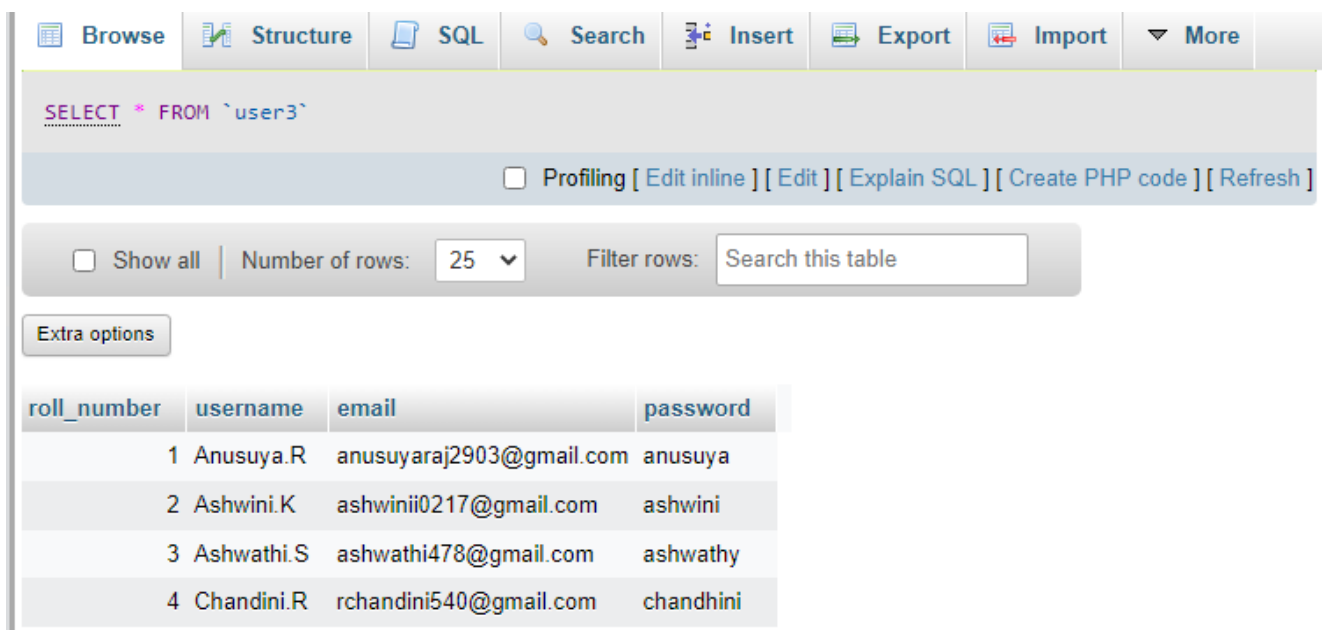
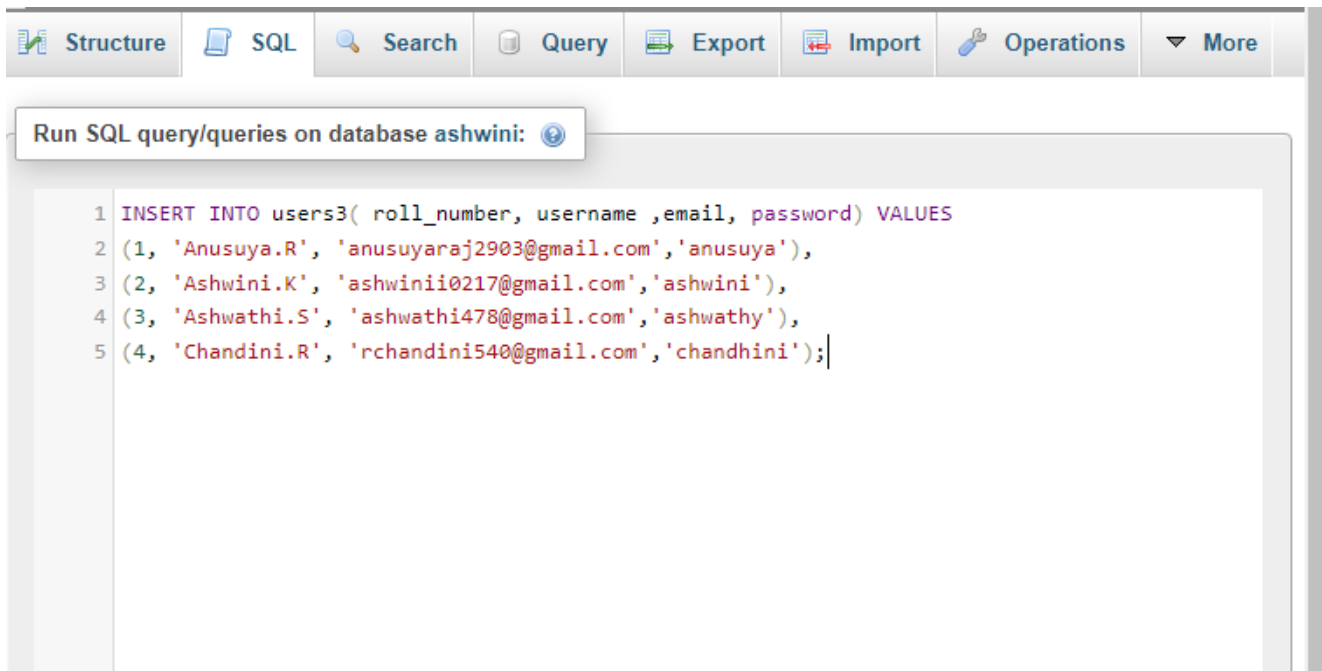
```
CREATE TABLE user3 (  
    roll_number int,  
    username varchar(300),  
    email varchar(300),  
    password varchar(300)  
);
```



1. Perform UPDATE, DELETE Queries with user table

INSERT Statement:

```
INSERT INTO user3( roll_number, username ,email, password) VALUES(1,
'Anusuya.R', 'anusuyaraj2903@gmail.com','anusuya'),
(2, 'Ashwini.K', 'ashwinii0217@gmail.com','ashu'),
(3, 'Ashwathi.S', 'ashwathi478@gmail.com','ashu07'),
(4, 'Chandini.R', 'rchandini540@gmail.com','chandinir');
```



UPDATE Statement:

UPDATE user3 SET username = 'Ashwini.K' WHERE roll_number = '2';

Browse

Structure

SQL

Search

Insert

Export

Import

More

Run SQL query/queries on table ashwini.user3: ?

```
1 UPDATE users4 SET username = 'Ashwini.K' WHERE roll_number = '2';
```

roll_number

username

email

password

Browse

Structure

SQL

Search

Insert

Export

Import

More

Show query box


✓ 0 rows affected. (Query took 0.0007 seconds.)


```
UPDATE user3 SET username = 'Ashwini.K' WHERE roll_number = '2';
```


[Edit inline] [Edit] [Create PHP code]


Insert Statement:


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


 Browse


 Structure


 SQL

 Search

 Insert

 Export

 Import

 More

Show query box


 1 row inserted. (Query took 0.0004 seconds.)


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


[\[Edit inline \]](#) [\[Edit \]](#) [\[Create PHP code \]](#)


DELETE Statement:


DELETE FROM user3 where roll_number='5'


 Browse


 Structure


 SQL


 Search

 Insert

 Export

 Import

 More

Run SQL query/queries on table ashwini.user3: 

1 DELETE FROM user3 where roll_number='5'|

roll_number
username
email
password

[Browse](#)
[Structure](#)
[SQL](#)
[Search](#)
[Insert](#)
[Export](#)
[Import](#)
[More](#)

⚠ Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

✓ Showing rows 0 - 3 (4 total, Query took 0.0003 seconds.)

SELECT * FROM `user3`

☐ Profiling
 [\[Edit inline \]](#)
[\[Edit \]](#)
[\[Explain SQL \]](#)
[\[Create PHP code \]](#)
[\[Refresh \]](#)

☐ Show all
 Number of rows: 25
 Filter rows:

Extra options

roll_number	username	email	password
1	Anusuya.R	anusuyaraj2903@gmail.com	anusuya
2	Ashwini.K	ashwinii0217@gmail.com	ashwini
3	Ashwathi.S	ashwathi478@gmail.com	ashwathy
4	Chandini.R	rchandini540@gmail.com	chandhini

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_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)
```

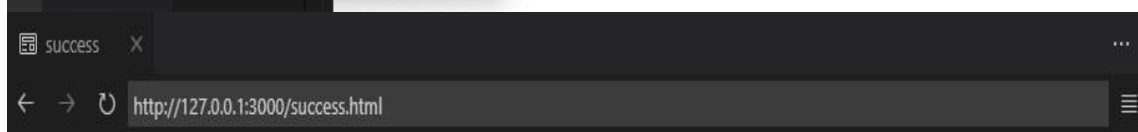
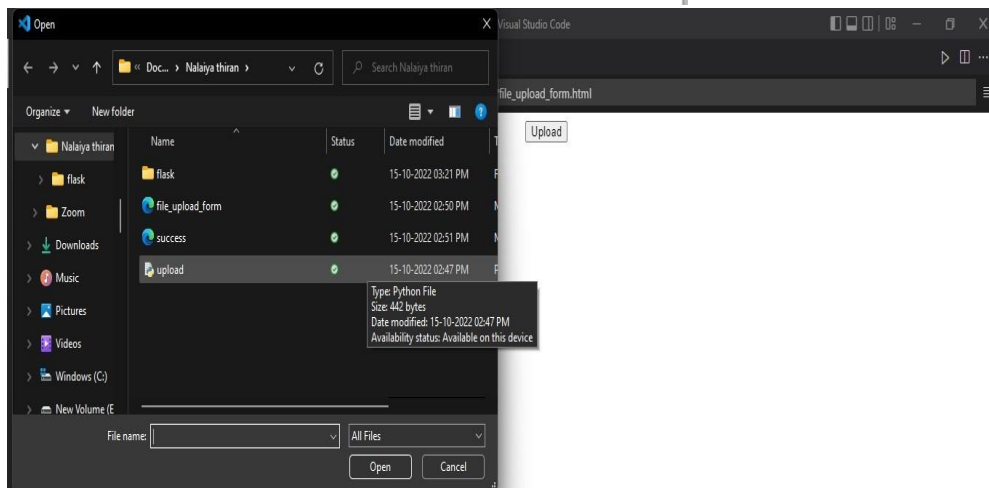
FILE_UPLOAD_FORM.HTML:

```
<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}}`