

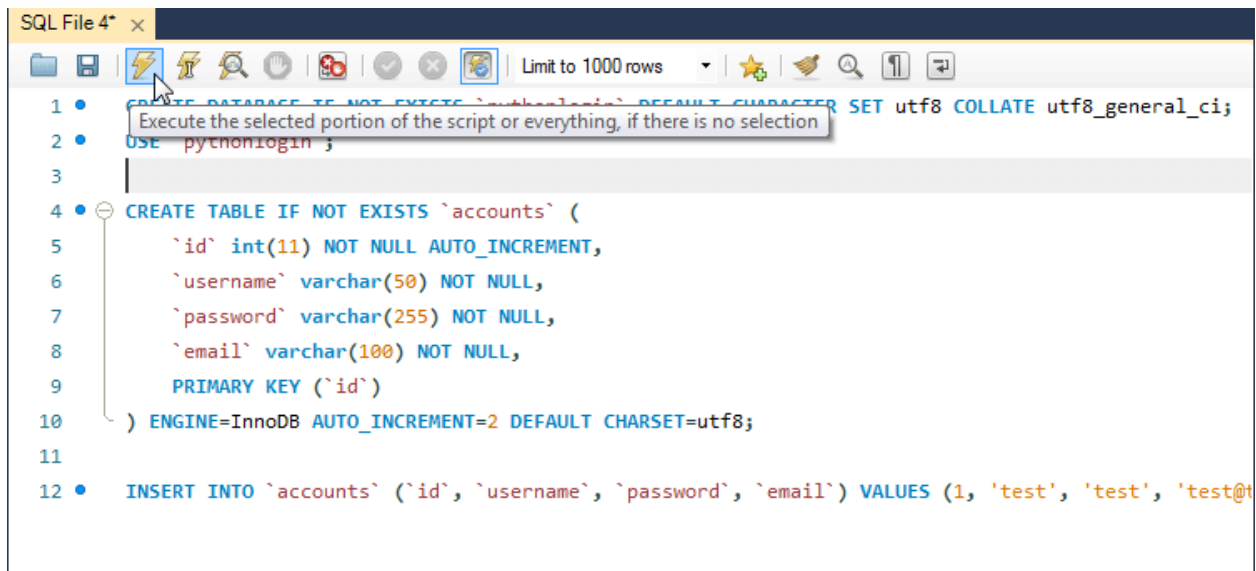
Assignment -2

Cloud Application Development Assignment Date	19 September 2022
Student Name	Ms. R.Jothika
Student Roll Number	61771921022
Maximum Marks	2 Marks

Question:

1. Create a flask App
2. Add the Home page, About page
3. Add the Bootstrap
4. Add the Sign in page and App the Signup page + database connectivity

Creating the Database and setting-up Tables:



The screenshot shows a SQL IDE window titled 'SQL File 4*'. The toolbar includes icons for file operations, execution, and search. A tooltip over the execution icon reads: 'Execute the selected portion of the script or everything, if there is no selection'. The SQL script contains the following commands:

```
1 • CREATE DATABASE IF NOT EXISTS `pythonlogin` DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci;
2 • USE `pythonlogin`;
3
4 • CREATE TABLE IF NOT EXISTS `accounts` (
5     `id` int(11) NOT NULL AUTO_INCREMENT,
6     `username` varchar(50) NOT NULL,
7     `password` varchar(255) NOT NULL,
8     `email` varchar(100) NOT NULL,
9     PRIMARY KEY (`id`)
10 ) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8;
11
12 • INSERT INTO `accounts` (`id`, `username`, `password`, `email`) VALUES (1, 'test', 'test', 'test@t
```

main.py

import the packages

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

from flask_mysqlldb import MySQL

import MySQLdb.cursors

import re

MySQL connection

```
app = Flask(__name__)
app.secret_key = 'your secret key'
app.config['MYSQL_HOST'] = 'localhost'
app.config['MYSQL_USER'] = 'root'
app.config['MYSQL_PASSWORD'] = ''
app.config['MYSQL_DB'] = 'pythonlogin'
mysql = MySQL(app)

@app.route('/pythonlogin/', methods=['GET', 'POST'])
def login():
    msg = ""
    return render_template('index.html', msg="")
```

Authenticating with users

```
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']
        return 'Logged in successfully!'
    else:
        msg = 'Incorrect username/password!'
```

Logout script

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

```
def logout():
```

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

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

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

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

Registering user

```
@app.route('/pythonlogin/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'^[^\s@]+@[^\s@]+\.[^\s@]+', 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)
```

Creating Home Page

```
@app.route('/pythonlogin/home')

def home():

    if 'loggedin' in session:

        return render_template('home.html', username=session['username'])

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

Creating profile page

```
@app.route('/pythonlogin/profile')

def profile():

    if 'loggedin' in session:

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

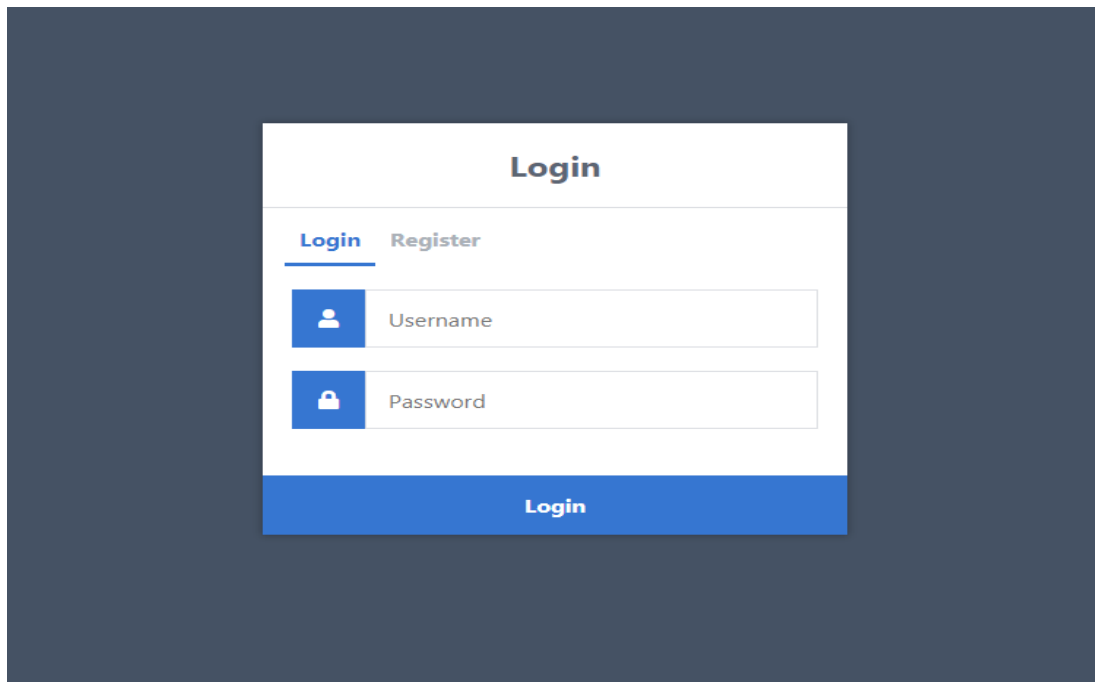
        cursor.execute('SELECT * FROM accounts WHERE id = %s', (session['id'],))

        account = cursor.fetchone()

        return render_template('profile.html', account=account)

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


Login Page:




The login page features a dark blue background with a white login form centered. The form has a title 'Login' at the top. Below the title are two tabs: 'Login' (active, underlined) and 'Register'. The form contains two input fields: 'Username' with a user icon and 'Password' with a lock icon. A blue 'Login' button is at the bottom.

Login

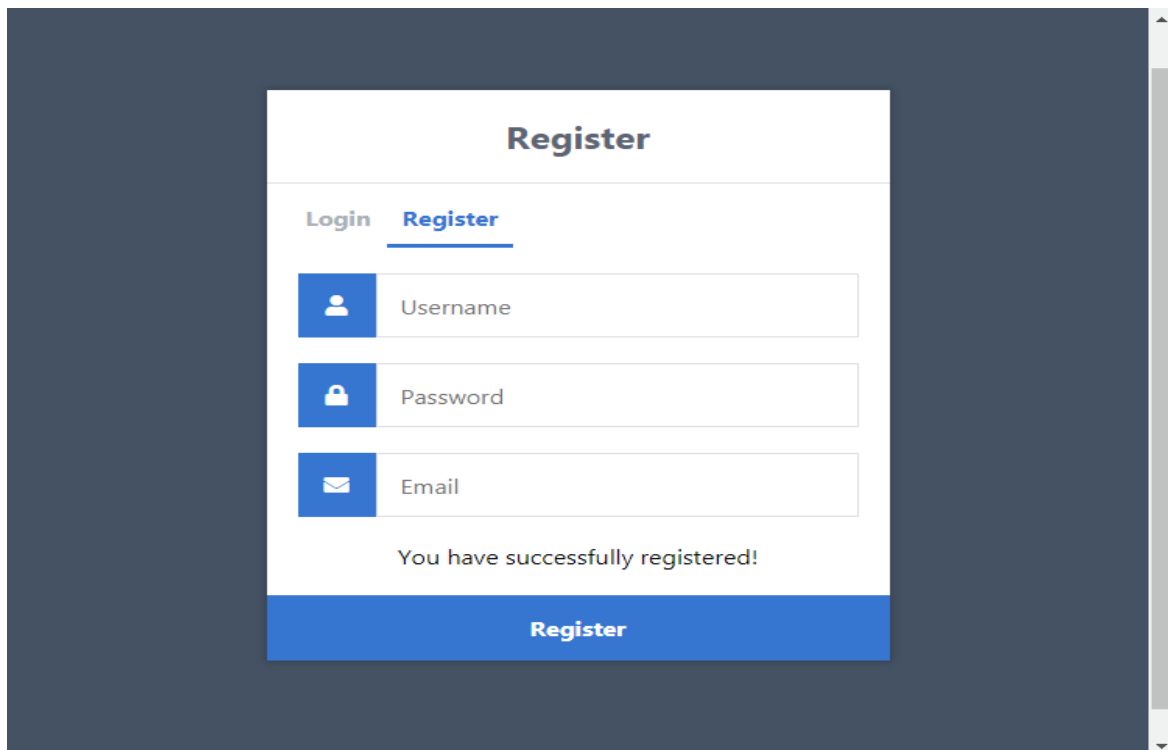
[Login](#) [Register](#)





Login


Register page:





The register page features a dark blue background with a white register form centered. The form has a title 'Register' at the top. Below the title are two tabs: 'Login' and 'Register' (active, underlined). The form contains three input fields: 'Username' with a user icon, 'Password' with a lock icon, and 'Email' with an envelope icon. A success message 'You have successfully registered!' is displayed below the fields. A blue 'Register' button is at the bottom.

Register

[Login](#) [Register](#)



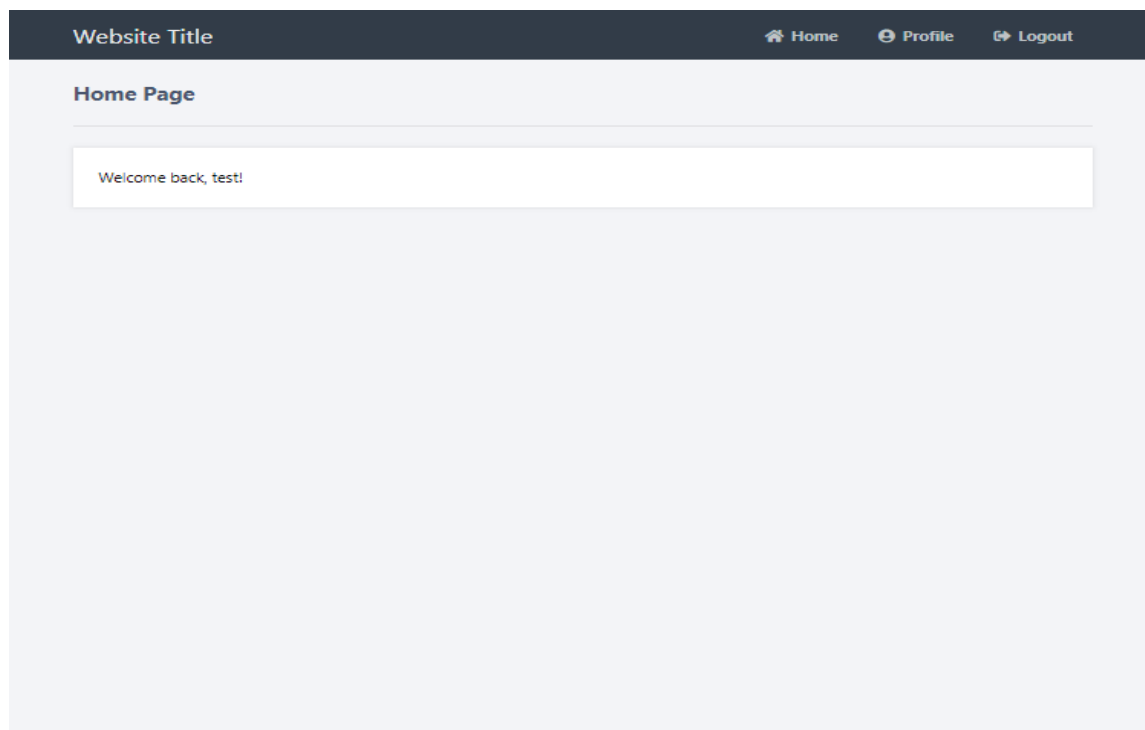




You have successfully registered!

Register

Home Page:



Profile Page:

