**Assignment -2**

## Python Programming

|  |  |
| --- | --- |
| Assignment Date | 04-11-2022 |
| Student Name | Lokeswar RR |
| Student Roll Number | 311119104040 |

1. Create a User table with Username, email, roll number, password

## Perform UPDATE,DELETE Queries with user table

1. Connect python code to db2.

## Create a flask app with registration page, login page and welcome page. By default load the registration page once the user enters all the fields store the data in database and navigate to login page authenticate user username and password. If the user is valid show the welcome page

Solution:

# App.py

from flask import Flask, render\_template, request, redirect

import sqlite3 as sql import models as dbHandler

app = Flask( name ) app.secret\_key = 'fasdgfdgdfg'

@app.route('/') def home():

return render\_template('home.html')

@app.route('/adduser') def new\_user():

return render\_template('add\_user.html')

@app.route('/addrec',methods = ['POST', 'GET']) def addrec():

if request.method == 'POST': try:

email = request.form['email'] un = request.form['username']

rn = request.form['rollnumber'] pin = request.form['pin']

with sql.connect("User\_database.db") as con:

cur = con.cursor()

cur.execute("INSERT INTO users (email,username,rollnumber,pin) VALUES (?,?,?,?)",(email,un,rn,pin) )

con.commit()

msg = "Record successfully added!" except:

con.rollback()

msg = "error in insert operation"

finally:

return render\_template("list.html",msg = msg) con.close()

@app.route('/list') def list():

con = sql.connect("User\_database.db") con.row\_factory = sql.Row

cur = con.cursor() cur.execute("select \* from users")

users = cur.fetchall()

return render\_template("list.html", users = users)

if name == ' main ':

app.run(debug = True)

@app.route("/delete") def delete():

return render\_template("delete.html")

@app.route('/deleterecord',methods = ["POST"]) def deleterecord():

un = request.form['username']

with sql.connect("User\_database.db") as con: try:

cur = con.cursor()

cur.execute("DELETE FROM users WHERE username = ?",[un]) con.commit()

msg = "Record successfully deleted" except:

msg = "can't be deleted" finally:

return render\_template("home1.html",msg = msg)

if name == ' main ':

app.run(debug = True)

@app.route('/deldb', methods = ["POST"]) def deldb():

con = sql.connect('User\_database.db') cur = con.cursor() cur.execute('DELETE FROM users;') con.commit()

con.close()

msg = 'All the data has been deleted'

return render\_template("home1.html",msg = msg)

@app.route("/log") def log():

return render\_template("login.html")

@app.route('/login', methods =['GET', 'POST']) def login():

un = request.form['username'] if request.method=='POST':

users = dbHandler.retrieveUsers() msg = 'Logged in successfully!'

return render\_template('welcome.html', users=un, msg=msg) else:

msg = 'You are not registered, would you like to be registered' return render\_template('home1.html', msg=msg)

if name == ' main ': app.run(debug=False, host='0.0.0.0')

# Models.py

**import sqlite3 as sql**

def retrieveUsers():

con = sql.connect("User\_database.db") cur = con.cursor()

cur.execute("SELECT username, pin FROM users") users = cur.fetchone()

con.close() return users

# sqlite\_db\_setup.py

importsqlite3

conn = sqlite3.connect('User\_database.db') print("Opened database successfully")

conn.execute('CREATE TABLE users (email TEXT, username TEXT, rollnumber INTEGER, pin INTEGER)')

print("Table created successfully") conn.close()

# Home.html

**<h1>Welcome to User DB APP</h1><br><br>**

<a href="/">HOME</a><br><br>

<a href="/adduser">User Registration</a><br><br>

<a href="/list">List User</a><br><br>

<a href="/log">Log in</a><br><br>

<a href="/delete">Remove a User</a>

# Add\_user.html

**<form action = "{{ url\_for('addrec') }}" method = "POST">**

<h3>User Information</h3> E-mail<br>

<input type = "email" name = "email" /></br>

Username<br>

<input type = "text" name = "username" /></br>

Rollnumber<br>

<input type = "text" name = "rollnumber" /><br>

PIN<br>

<input type = "password" name = "pin" min="4" max="8" /><br><br>

<input type = "submit" value = "submit" /><p> </p>

<input type = "reset"/>

</form>

# list.html

**<!doctype html>**

<html>

<body>

<a href="/">HOME</a><br><br>

<a href="/adduser">Add New Student</a><br><br>

<a href="/list">List Student</a><br><br>

<br><hr>

{{ msg }}

<table border = 1>

<thead>

<td> Email </td>

<td> Username </td>

<td> Roll Number </td>

<td> Pin </td>

</thead>

{% for row in users %}

<tr>

<td>{{row["email"]}}</td>

<td>{{row["username"]}}</td>

<td> {{ row["rollnumber"]}}</td>

<td>{{row['pin']}}</td>

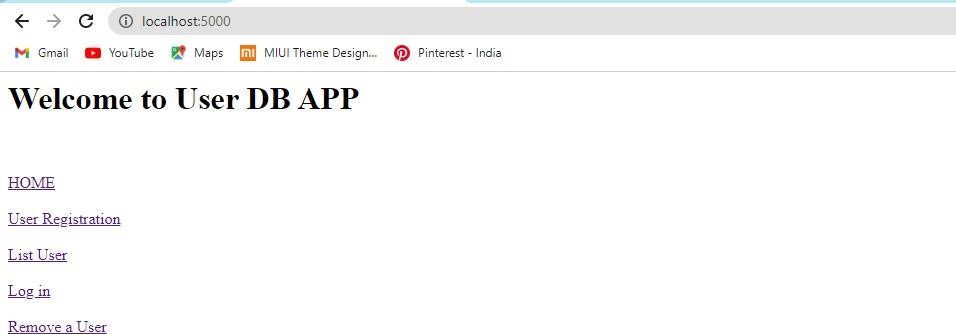
</tr>

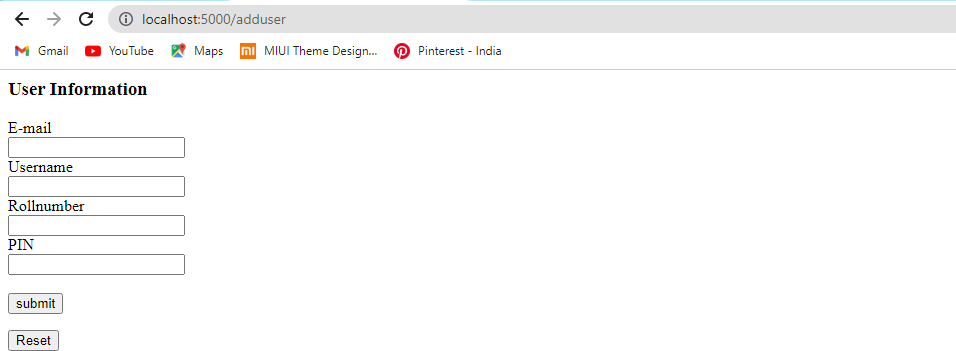
{% endfor %}

</table>

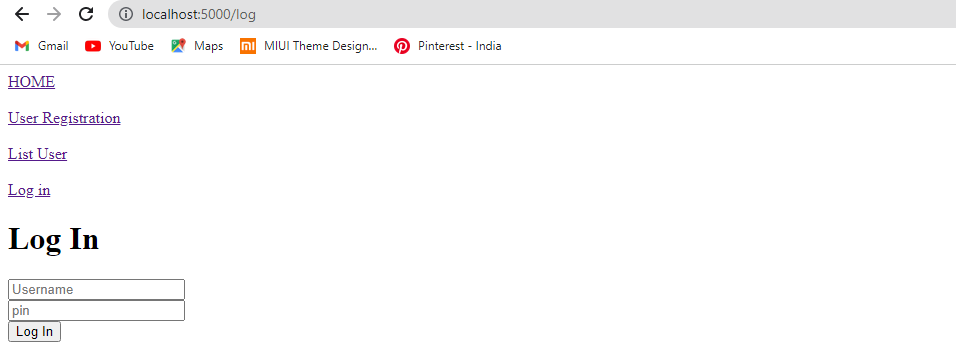
</body>

</html>

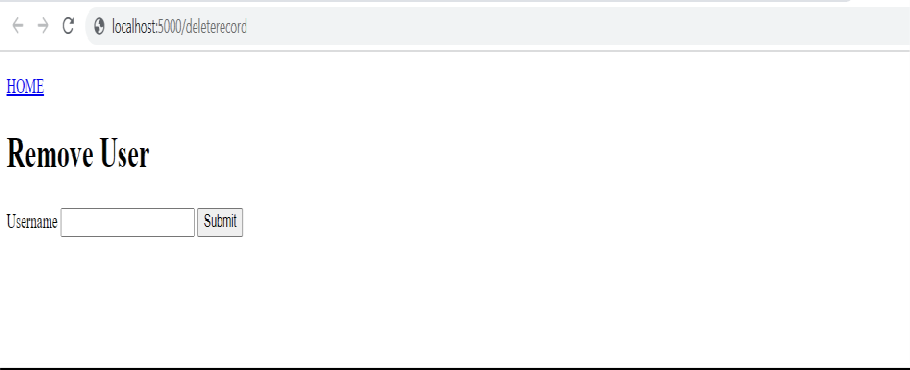




|  |  |  |  |
| --- | --- | --- | --- |
| **EMAIL** | **USER NAME** | **ROLL NUMBER** | **PIN** |
| [sanjay.23cs@licet.ac.in](mailto:sanjay.23cs@licet.ac.in) | Sanjay K | 311119104046 | 9876 |
| [lokeswar.23cs@licet.ac.in](mailto:lokeswar.23cs@licet.ac.in) | Lokeswar | 311119104040 | 1234 |
| [melvynfrancis.23cs@licet.ac.in](mailto:melvynfrancis.23cs@licet.ac.in) | Melvyn Francis | 311119104045 | 3345 |
| [melwinjones.23cs@licet.ac.in](mailto:melwinjones.23cs@licet.ac.in) | Melwin Jones J | 311119104046 | 5412 |









|  |  |  |  |
| --- | --- | --- | --- |
| **EMAIL** | **USER NAME** | **ROLL NUMBER** | **PIN** |
| [sanjay.23cs@licet.ac.in](mailto:sanjay.23cs@licet.ac.in) | Sanjay K | 311119104046 | 9876 |
| [lokeswar.23cs@licet.ac.in](mailto:lokeswar.23cs@licet.ac.in) | Lokeswar | 311119104040 | 1234 |
| [melvynfrancis.23cs@licet.ac.in](mailto:melvynfrancis.23cs@licet.ac.in) | Melvyn Francis | 311119104045 | 3345 |