**Assignment -2**

Python Programming

| | |
|---|---|
| Assignment Date | 19 September 2022 |
| Student Name | Lokhitha D |
| Student Roll Number | 211419205101 |
| Maximum Marks | 2 Marks |

**Question-1:**

1. Create user table with user with email ,username, roll number, password.2. Perform UPDATE,DELETE Queries with user table 3. Connect python code to db2. 4. 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:**

```
from flask import Flask, render_template, request,
```

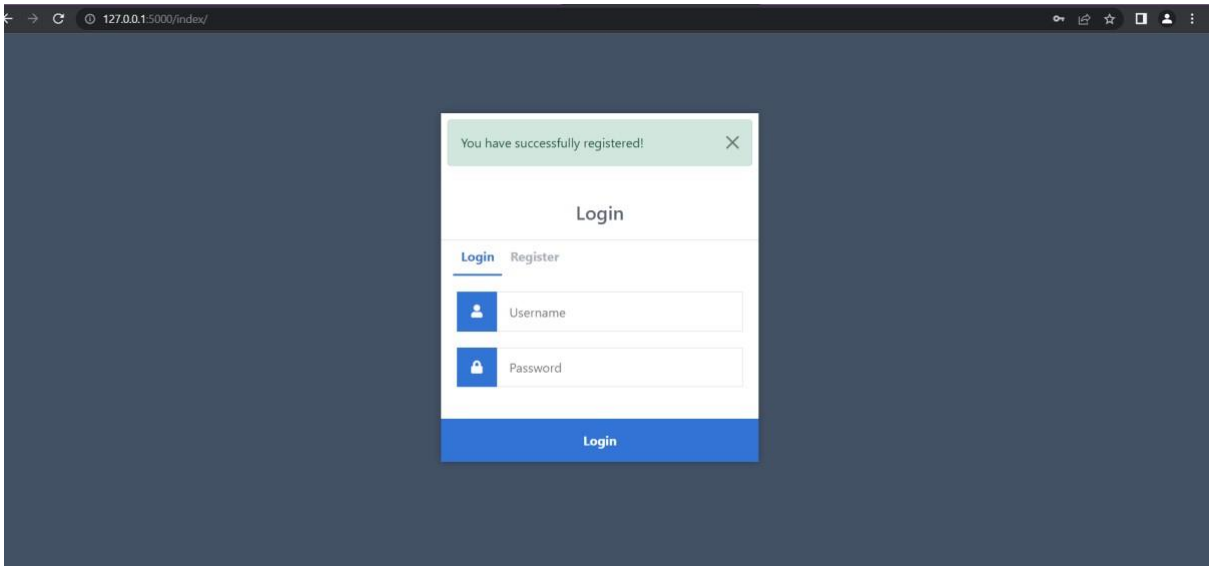
```
redirect, url_for, session,flash from
flask_mysqldb import MySQL
 import mysql.connector as
mysql

import re


app = Flask(__name__)

# Change this to your secret key (can be anything,
it's  for  extra  protection)  app.secret_key  =
'1a2b3c4d5e'

# Enter your database connection details below
app.config['MYSQL_HOST'] = 'localhost'
app.config['MYSQL_PORT'] = 3306
app.config['MYSQL_USER'] = 'root'
app.config['MYSQL_PASSWORD'] = 'root'
#app.config['MYSQL_CURSORCLASS'] = 'DictCursor'
app.config['MYSQL_DB'] = 'flaskproject'

#app.config['MYSQL_PORT'] = '3306'

# Intialize MySQL mysql
= MySQL(app)
```

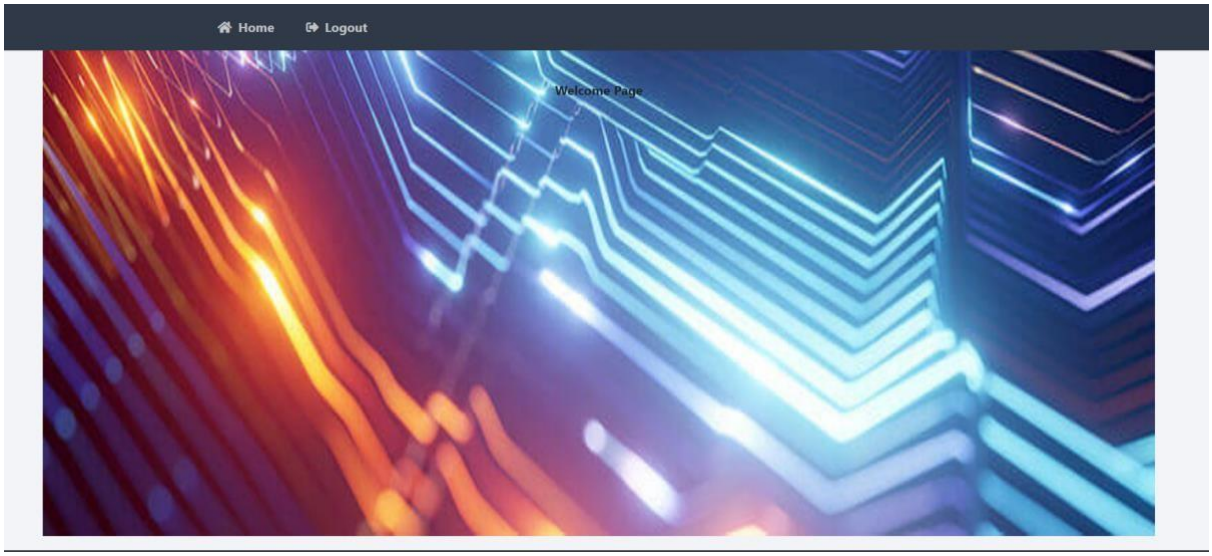
```
# http://localhost:5000/pythonlogin/ - this will be
the login page, we need to use both GET and POST
requests
@app.route('/index/', methods=['GET', 'POST']) def
login():
# Output message if something goes wrong...
    # Check if "username" and "password" POST
requests exist (user submitted form)     if
request.method == 'POST' and 'username' in
request.form and 'password' in request.form:
# Create variables for easy access         username =
request.form['username']         password =
request.form['password']         # Check if account
exists using MySQL         cursor =
mysql.connection.cursor()
        #cursor =
mysql.connection.cursor(MySQLdb.cursors.DictCursor)
        #cursor =
mysql.connection.cursor(MySQLdb.cursors.DictCursor)
cursor.execute('SELECT * FROM accounts WHERE username
= %s AND password = %s', (username, password))

        # Fetch one record and return result
account = cursor.fetchone()
                # If account exists in accounts
table in out database        # if account:
if request.method == 'POST':
            # Create session data, we can access
this data in other routes
session['loggedin'] = True
#session['id'] = account['id']
session['username'] = request.form['username']
            #session['password'] = account['password']
            # Redirect to home page
return redirect(url_for('home'))
else:
            # Account doesnt exist or
username/password incorrect
flash("Incorrect username/password!",
"danger")     return
render_template('auth/login.html',title="Login")



# http://localhost:5000/pythinlogin/register
```

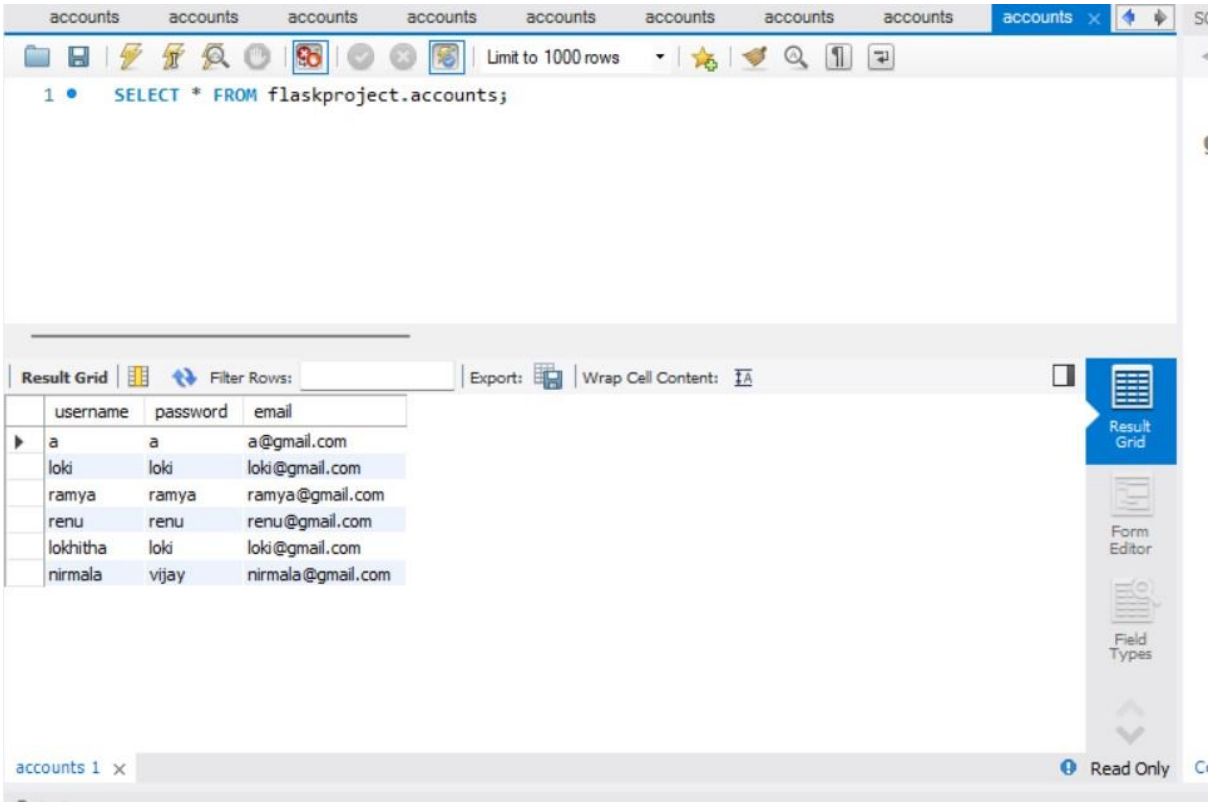
```
# This will be the registration page, we need to use
both GET and POST requests
@app.route('/pythonlogin/register', methods=['GET',
'POST']) def
register():
    # Check if "username", "password" and "email"
POST requests exist (user submitted form)     if
request.method == 'POST' and 'username' in
request.form and 'password' in request.form and
'email' in request.form:
        # Create variables for easy access
username = request.form['username']
password = request.form['password']         email
= request.form['email']
                # Check if account exists using MySQL
cursor = mysql.connection.cursor()
        # cursor.execute('SELECT * FROM accounts
WHERE username = %s', (username))
cursor.execute( "SELECT * FROM accounts WHERE
username LIKE %s", [username] )         account =
cursor.fetchone()
        # If account exists show error and validation
checks         if account:
            flash("Account already exists!", "danger")
elif not re.match(r'[^@]+@[^@]+\.[^@]+', email):
            flash("Invalid email address!", "danger")
elif not re.match(r'[A-Za-z0-9]+', username):
            flash("Username must contain only
characters and numbers!", "danger")         elif
not username or not password or not email:
            flash("Incorrect username/password!",
"danger")
else:
        # Account doesnt exists and the form data is
valid, now insert new account into accounts table
cursor.execute('INSERT INTO accounts VALUES (%s, %s,
%s)', (username,password, email))
mysql.connection.commit()             flash("You have
successfully registered!",
"success")             return
redirect(url_for('login'))     elif request.method
== 'POST':         # Form is empty... (no POST
data)         flash("Please fill out the form!",
"danger")     # Show registration form with message
(if any)
```

```
    return
render_template('auth/register.html',title="Register")

# http://localhost:5000/pythinlogin/home
# This will be the home page, only accessible for
loggedin users

@app.route('/pythonlogin/home') def
home():
    # Check if user is loggedin
if 'loggedin' in session:
        # User is loggedin show them the home page
return render_template('home/home.html',
username=session['username'],title="Home")     #
User is not loggedin redirect to login page
return redirect(url_for('login'))


@app.route('/pythonlogin/profile') def
profile():
    # Check if user is loggedin
if 'loggedin' in session:
        # User is loggedin show them the home page
return render_template('auth/profile.html',
username=session['username'],title="Profile")     #
User is not loggedin redirect to login page
return redirect(url_for('login'))
 if __name__
=='__main__':
app.run(debug=True)
```

**OUTPUT:**


127.0.0.1:5000/index/
You have successfully registered!
Login
Login
Register
Username
Password
Login


Home
Logout
Welcome Page


accounts
accounts
accounts
accounts
accounts
accounts
accounts
accounts
accounts
Limit to 1000 rows
1 SELECT * FROM flaskproject.accounts;
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
username password email
a a a@gmail.com
loki loki loki@gmail.com
ramya ramya ramya@gmail.com
renu renu renu@gmail.com
lokhitha loki loki@gmail.com
nirmala vijay nirmala@gmail.com
Result Grid
Form Editor
Field Types
accounts 1
Read Only