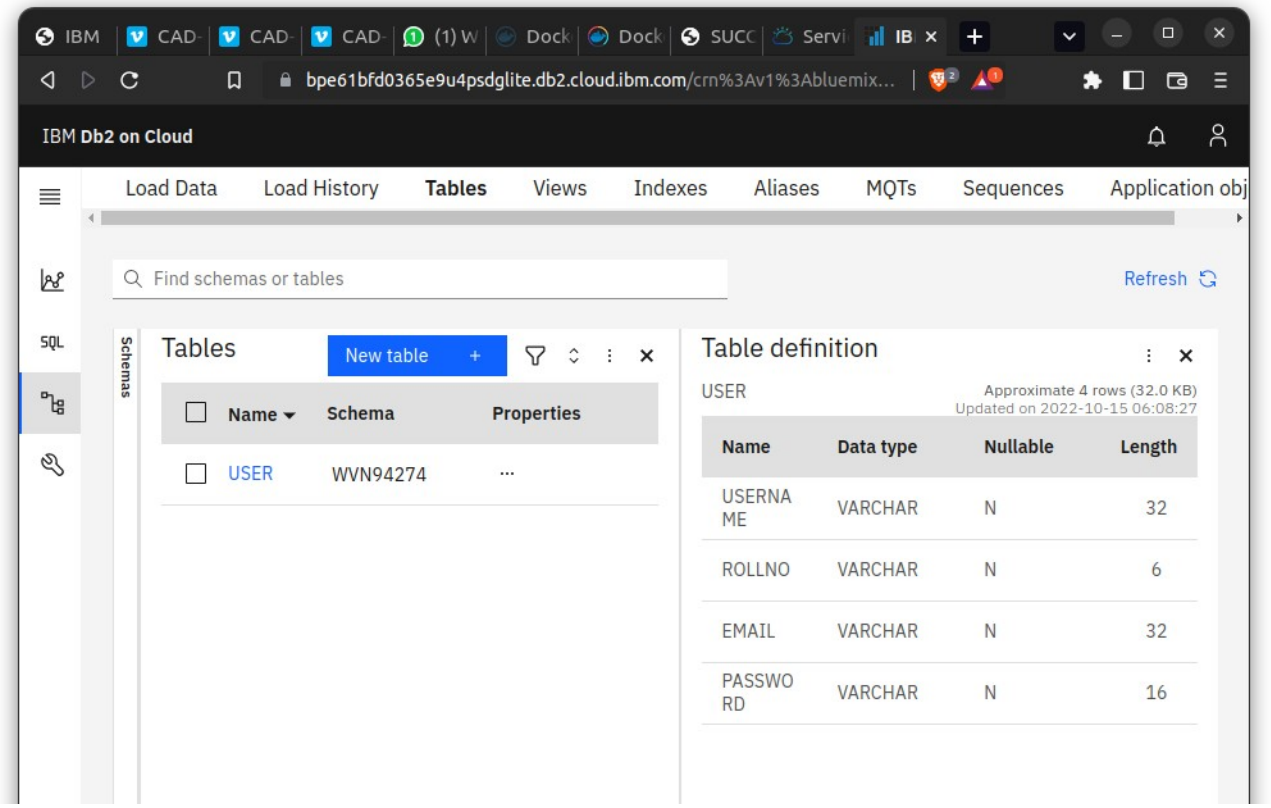
Assignment-2

1.Create user table with email,username and phone number and password.

DB2_QUERY:

create table if not exists user(username varchar(32) null,rollno varchar(6) null,email varchar(32) null,password varchar(16) null);

Output:



2.Perform UPDATE,DELETE Quries with user table.

DB2_QUERY:

--Inserting_into_table

insert into user values
('shagish',312323,'962219104095@gmail.com','helloworld'),
('benoj',342334,'962219205017@gmail.com','hellobruh'),
('shami',626327,'962219104097@gmail.com','hellodood')

(‘sharn’,’111956’,962219104100@gmail.com’,’sharnsharn’)
('Shahina',897343,'962219104098@gmail.com','haihai')
('sreeja',897234,'962219104109@gmail.com','hihihi');

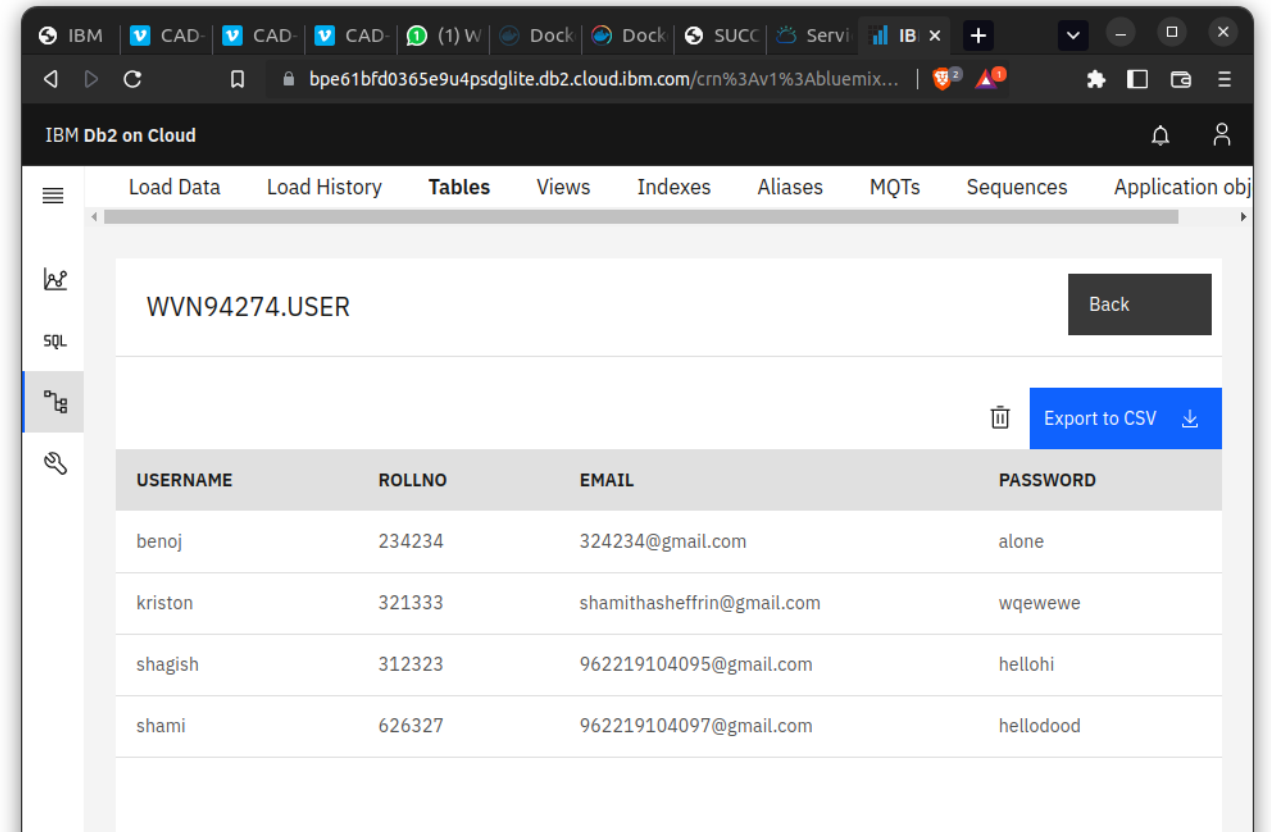
--Updating_Table_Values

update user set password='hellohi' where username='shagish';

--Deleting_table_rows

delete from user where username='sharn';

Output:



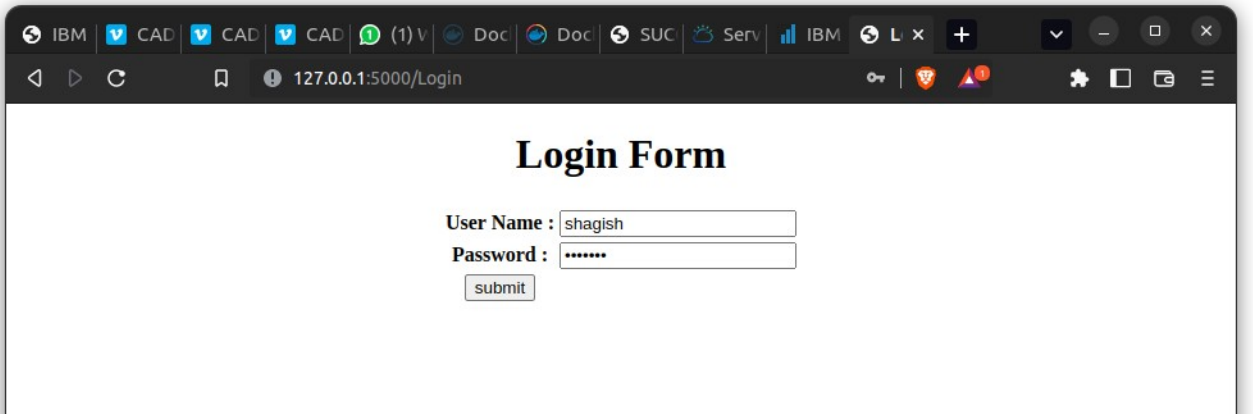
3.Connect python code to DB2
4.Create a flask app with registration page,login page and welcome page .By deafult load the registration page once user enters all the fields store the data in data base and navigate to login page authenticate user username and password .If the user is valid show welcome page.

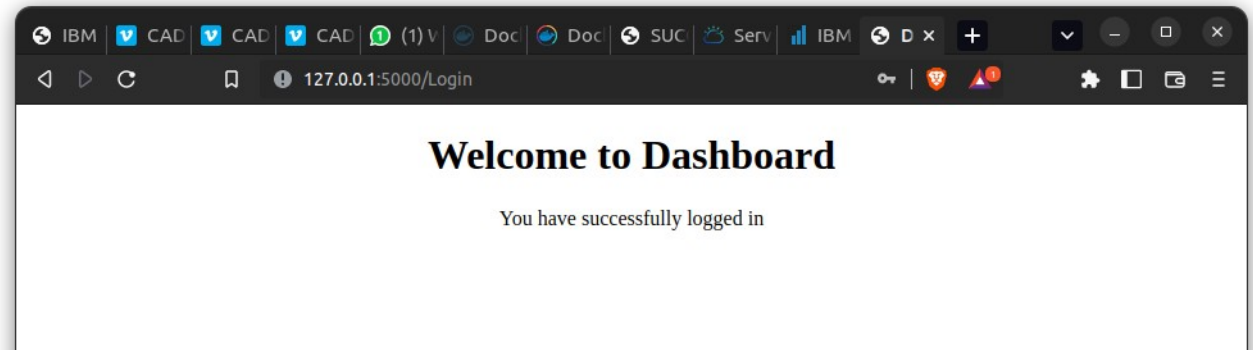
app.py

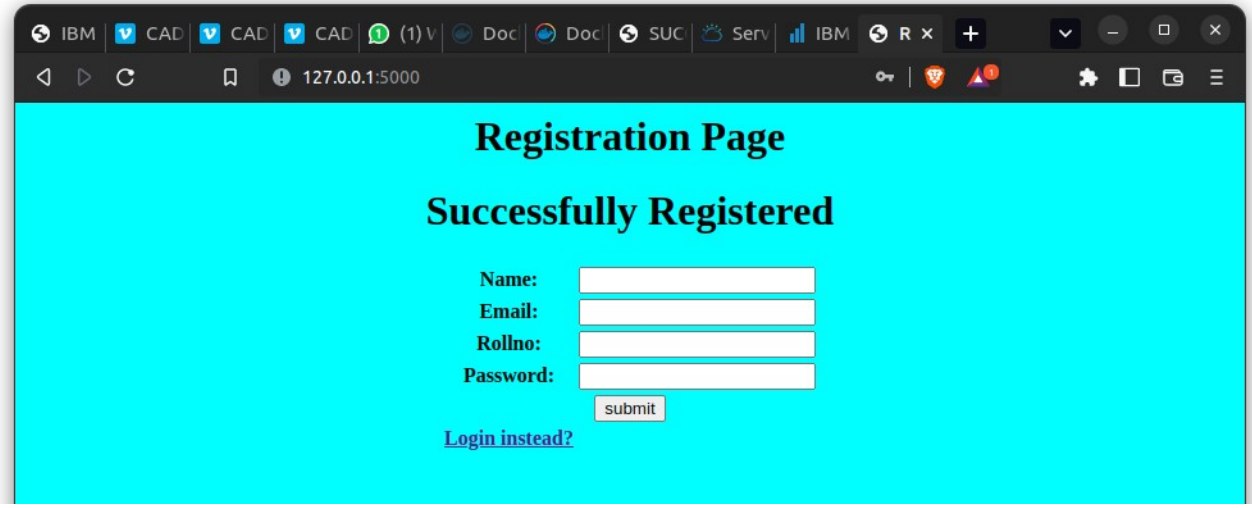
```
from flask import Flask, request, redirect, render_template
import ibm_db
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=125f9f61-9715-46f9-
9399-
c8177b21803b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=30426;SEC
URITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=wvn94274;PW
D=2K5Z7ZiQuEV2edmQ", '', '')


app = Flask(__name__)


@app.route('/', methods=['POST', 'GET'])
def register():
    if request.method=="POST":
        msg=''
        name = request.form.get('name')
        email = request.form.get('email')
        rollno = request.form.get('rollno')
        password = request.form.get('password')
        stmt = ibm_db.prepare(conn, 'SELECT * FROM user WHERE username=?')
        ibm_db.bind_param(stmt, 1, name)
        ibm_db.execute(stmt)
        rs = ibm_db.fetch_assoc(stmt)
        print(rs)
        if rs:
            msg = 'Account already Exists'
            return render_template('register.html', msg=msg)
        else:
            reg_stmt = ibm_db.prepare(conn, 'INSERT INTO user VALUES(?,?,?,?)')
            ibm_db.bind_param(reg_stmt, 1, name)
            ibm_db.bind_param(reg_stmt, 2, rollno)
            ibm_db.bind_param(reg_stmt, 3, email)
            ibm_db.bind_param(reg_stmt, 4, password)
            ibm_db.execute(reg_stmt)
            msg = 'Successfully Registered'
            return render_template('register.html', msg=msg)
    else:
        return render_template('register.html')
```

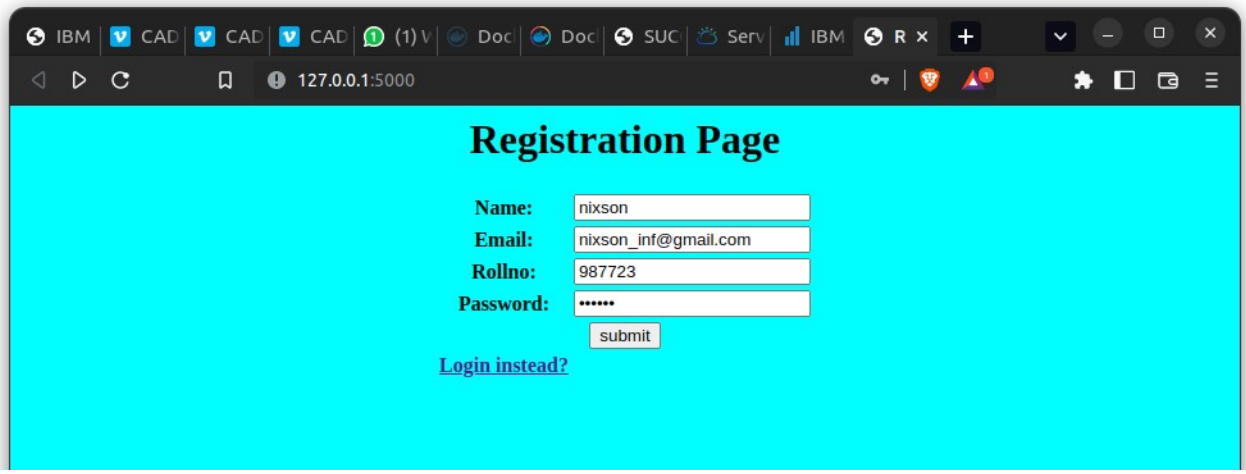
```
@app.route('/Login', methods=['POST', 'GET'])
def login():
    if request.method=="POST":
        name = request.form.get('name')
        password = request.form.get('password')
        log_stmt = ibm_db.prepare(conn, 'SELECT * FROM user WHERE username=?
and password=?')
        ibm_db.bind_param(log_stmt, 1, name)
        ibm_db.bind_param(log_stmt, 2, password)
        ibm_db.execute(log_stmt)
        rs = ibm_db.fetch_assoc(log_stmt)
        if rs:
            return render_template('dashboard.html')
        else:
            msg = 'UID/Password is incorrect'
            return render_template('login.html', msg=msg)
    else:
        return render_template('login.html')


@app.route('/dashboard')
def dashboard():
    return render_template('dashboard.html')

if __name__ == '__main__':
    app.run(debug=True)
```

Output:




127.0.0.1:5000/Login
Welcome to Dashboard
You have successfully logged in


127.0.0.1:5000
Registration Page
Successfully Registered
Name:
Email:
Rollno:
Password:
submit
Login instead?


127.0.0.1:5000
Registration Page
Name:
nixson
Email:
nixson_inf@gmail.com
Rollno:
987723
Password:
submit
Login instead?