

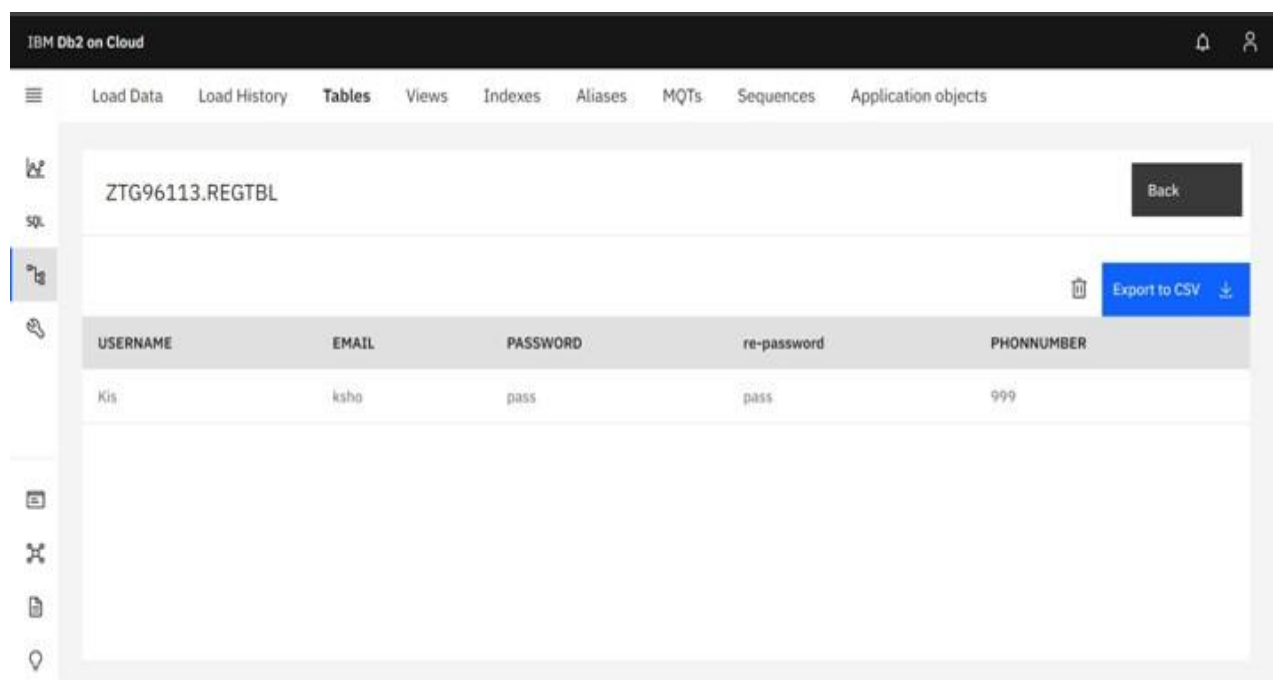
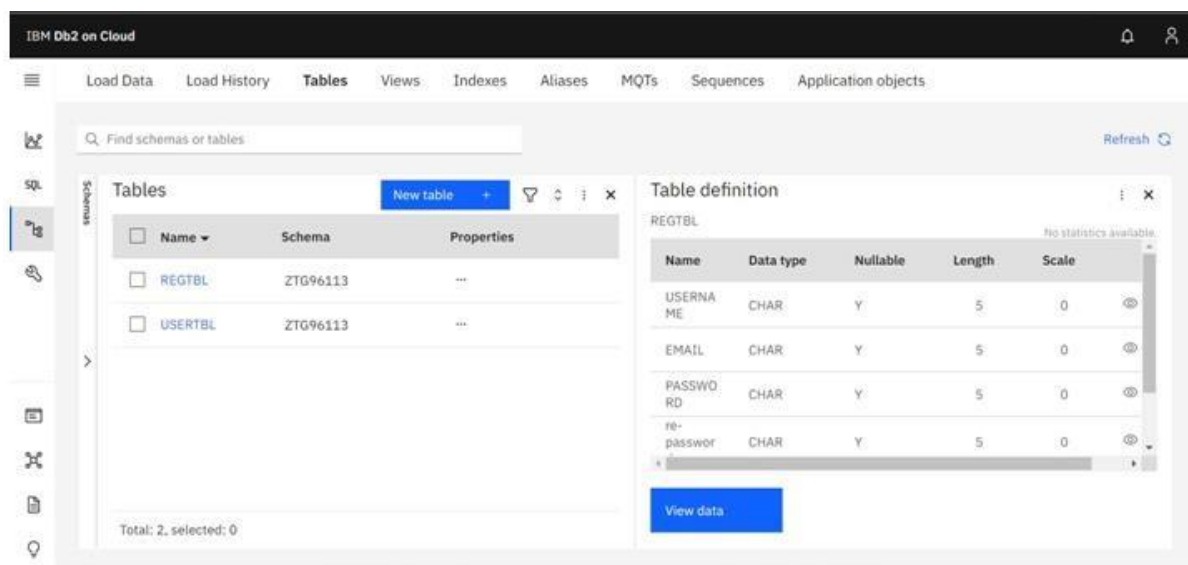
IMPLEMENTING WEB APPLICATION

Create IBM DB2 And Connect With Python

Team ID	PNT2022TMID32524
Project Name	Plasma Donor Application

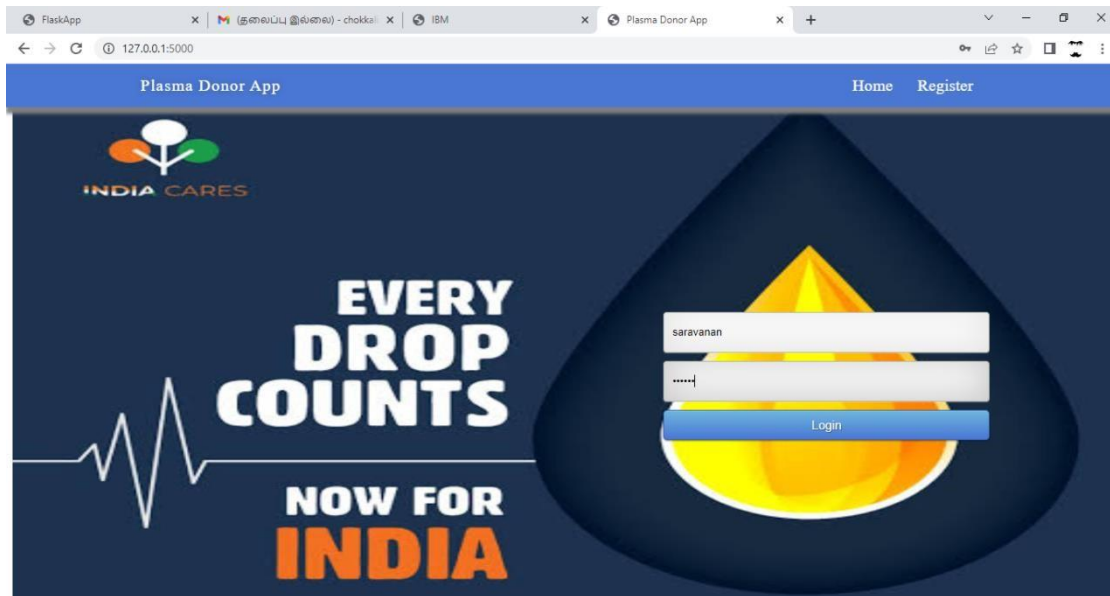
IBM DB2:

STEP 1: In IBM Db2 Create A table And Enter Table name ,column name,Row name, And Set Data Type



Login Page:

Step 2: Already Registered User Enter Their Name and Password To Login



Plasma Donor App

Home Register

INDIA CARES

EVERY
DROP
COUNTS

NOW FOR
INDIA

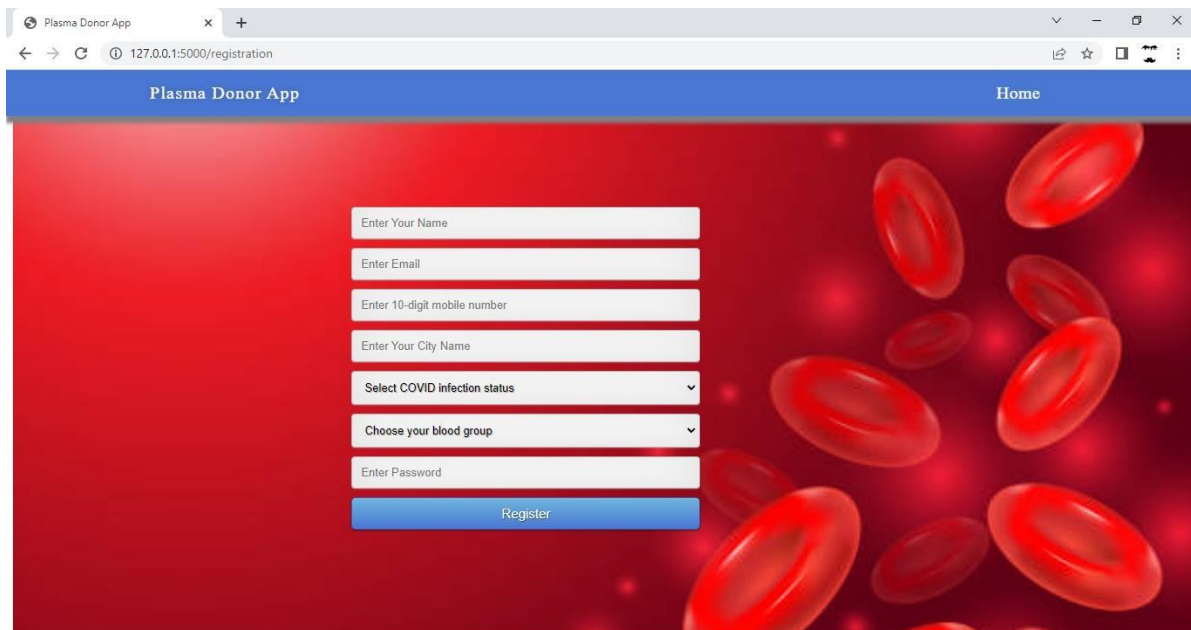
saravanan

.....

Login

Registration Page:

Step 3: New User Please Fill The Registration Form Details And Go To Login



Plasma Donor App

Home

Enter Your Name

Enter Email

Enter 10-digit mobile number

Enter Your City Name

Select COVID infection status

Choose your blood group

Enter Password

Register

```
from flask import Flask, render_template, request, redirect, url_for,
session import ibm_db import re

app = Flask( name )
```



```

app.secret_key = 'a'

conn = ibm_db.connect(

"DATABASE=#;HOSTNAME=#;PORT=32731;USERNAME=#PASSWORD=#;SECURITY=SSL;SSL
SERVERCERTIFICATE=DigiCertGlobalRootCA.crt;", "", "")

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

    msg = '' if request.method == 'POST': username =
    request.form['username'] email =
    request.form['email'] password =
    request.form['password'] sql = "SELECT * FROM
    users WHERE username =?" stmt =
    ibm_db.prepare(conn, sql) ibm_db.bind_param(stmt,
    1, username) ibm_db.execute(stmt) account =
    ibm_db.fetch_assoc(stmt) print(account) 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 = 'name must contain only characters and numbers !' else:

                insert_sql = "INSERT INTO users VALUES (?, ?, ?)"
                prep_stmt = ibm_db.prepare(conn, insert_sql)
                ibm_db.bind_param(prepare_stmt, 1, username)
                ibm_db.bind_param(prepare_stmt, 2, email)

```



```

        ibm_db.bind_param(prepare_stmt, 3, password)

        ibm_db.bind_param(prepare_stmt, 4, re-password)

        ibm_db.bind_param(prepare_stmt, 6, phonnumber)

        ibm_db.execute(prepare_stmt) msg = 'You have

        successfully registered !'

elif request.method == 'POST':

    msg = 'Please fill out the form !' return

render_template('register.html', msg=msg)

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

def login(): global userid msg = ''

    if request.method == 'POST':

        username = request.form['username'] password =

        request.form['password'] sql = "SELECT * FROM users WHERE

        username =? AND password=?" stmt = ibm_db.prepare(conn,

        sql) ibm_db.bind_param(stmt, 1, username)

        ibm_db.bind_param(stmt, 2, password) ibm_db.execute(stmt)

        account = ibm_db.fetch_assoc(stmt) print(account) if

        account:

            session['loggedin'] = True session['id']

            = account['USERNAME'] userid =

            account['USERNAME'] session['username'] =

            account['USERNAME'] msg = 'Logged in

            successfully !'

            msg = 'Logged in successfully !'

```

```
        return render_template('dashboard.html', msg=msg)

    else:

        msg = 'Incorrect username / password !'

    return render_template('login.html', msg=msg)


if __name__ == '__main__':
    app.run(host='0.0.0.0')
    # app.run(debug=True)
```