

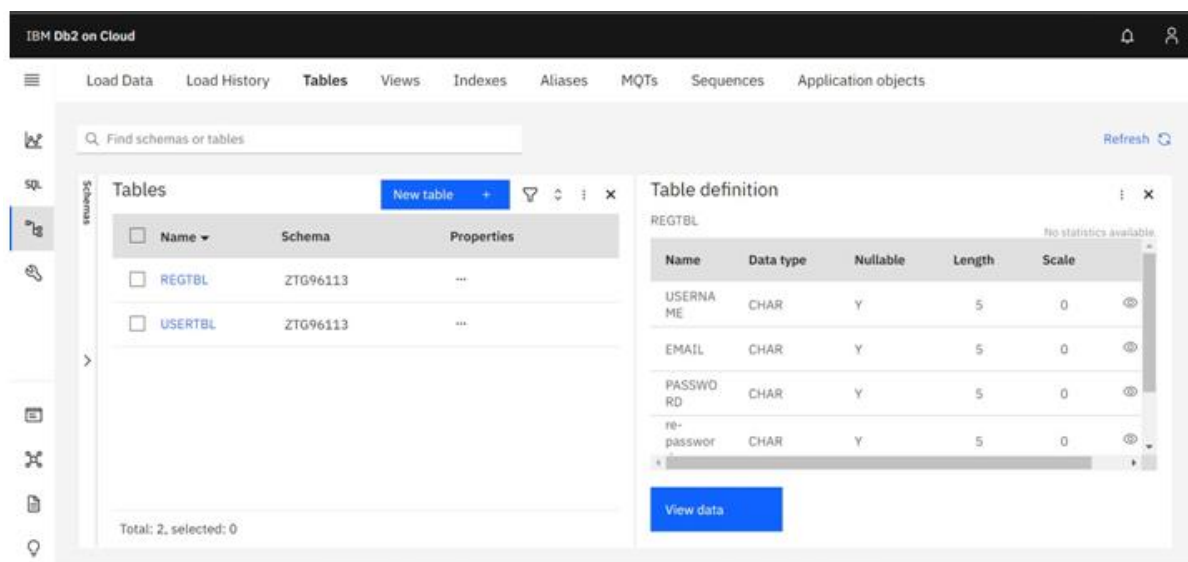
IMPLEMENTING WEB APPLICATION

Create IBM DB2 And Connect With Python

Team ID	PNT2022TMID49331
Project Name	Plasma Donor Application

IBM DB2:

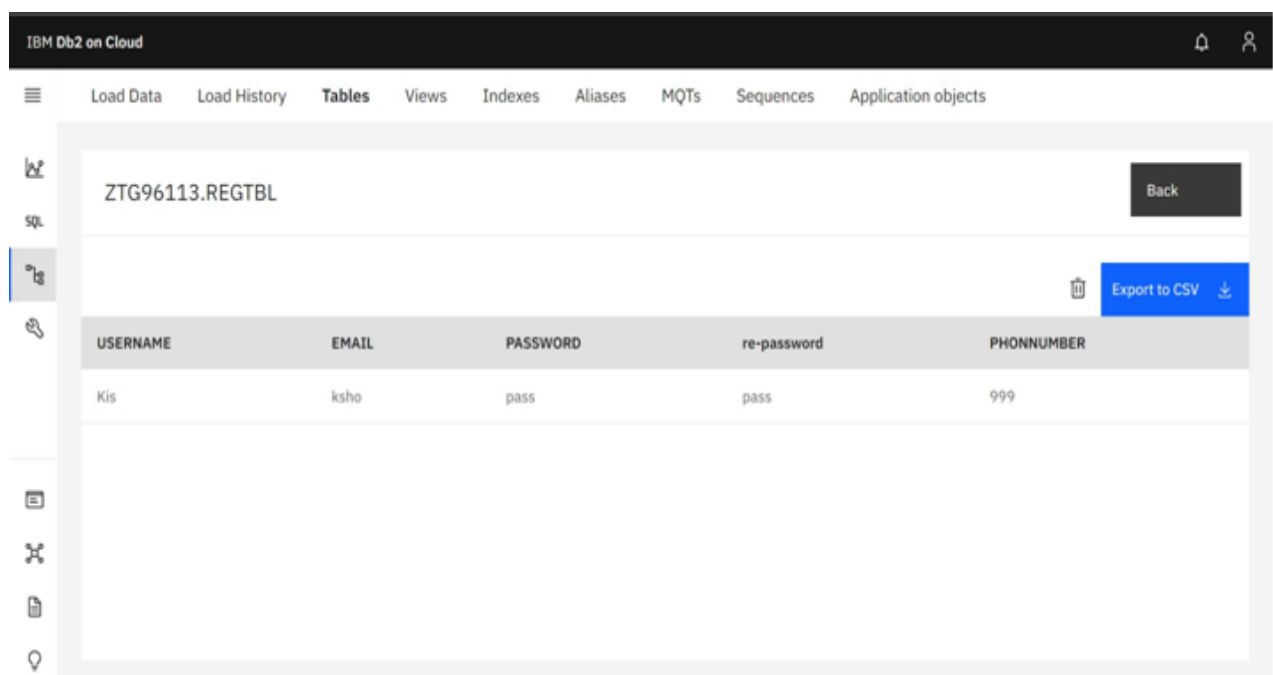
STEP 1: In IBM Db2 Create A table And Enter Table name ,column name,Row name, And Set Data Type



The screenshot shows the IBM Db2 on Cloud interface. The 'Tables' tab is selected, displaying a list of tables in the ZTG96113 schema: REGTBL and USERTBL. The 'Table definition' panel for REGTBL is open, showing the following columns:

Name	Data type	Nullable	Length	Scale
USERNAME	CHAR	Y	5	0
EMAIL	CHAR	Y	5	0
PASSWORD	CHAR	Y	5	0
re-password	CHAR	Y	5	0

A 'View data' button is visible at the bottom of the table definition panel.



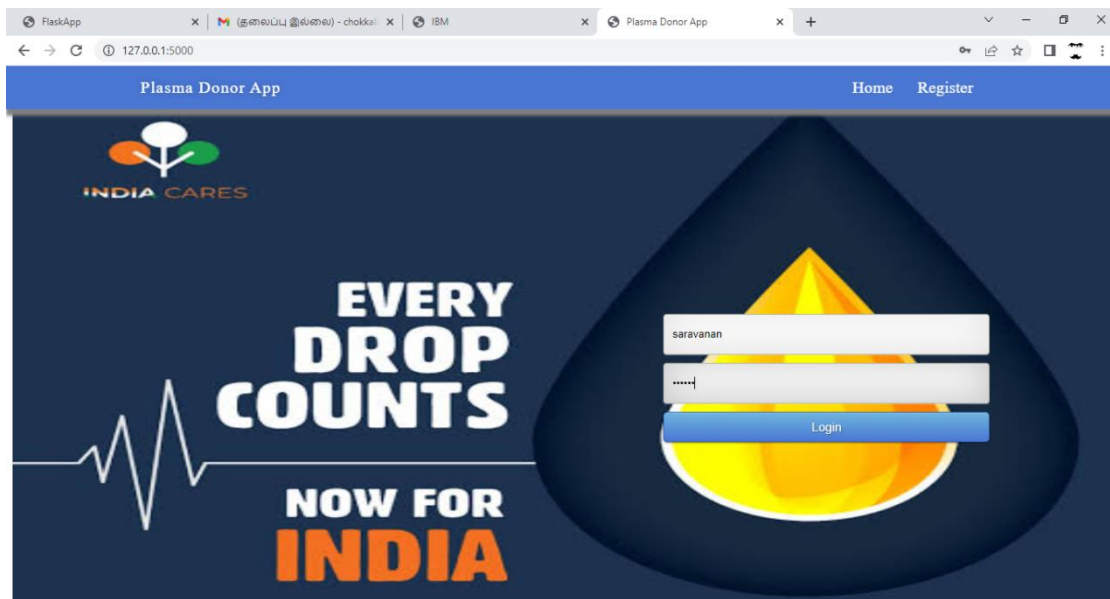
The screenshot shows the IBM Db2 on Cloud interface with the 'ZTG96113.REGTBL' table selected. The table data is displayed in a grid format:

USERNAME	EMAIL	PASSWORD	re-password	PHONNUMBER
Kis	ksho	pass	pass	999

An 'Export to CSV' button is visible in the top right corner of the table view.

Login Page:

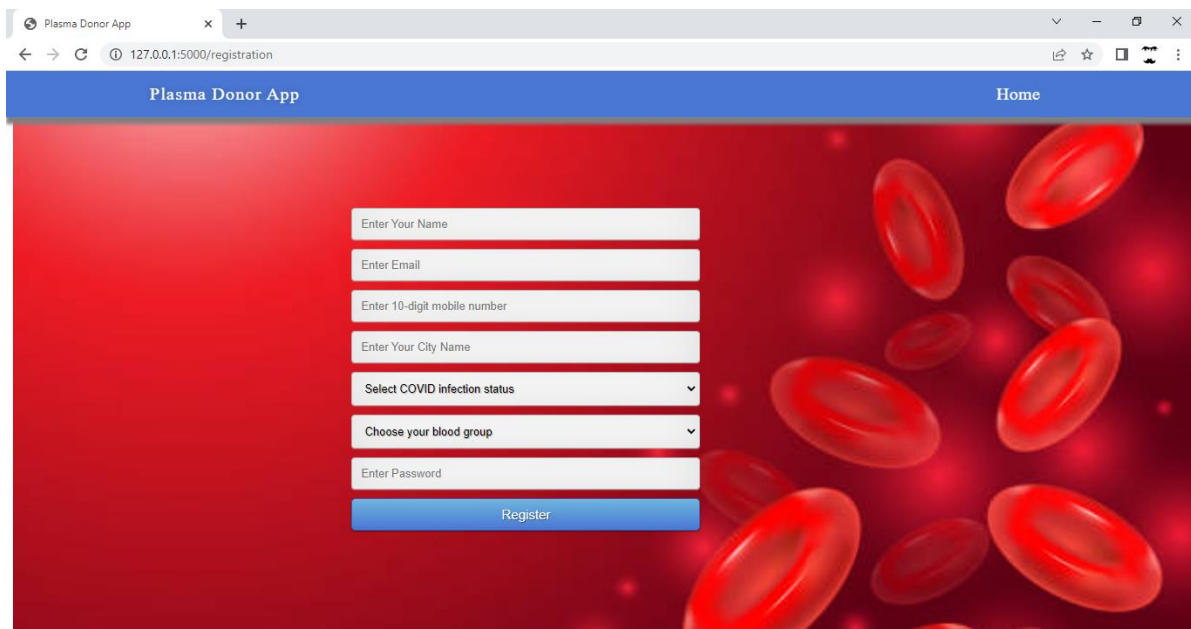
Step 2: Already Registered User Enter Their Name and Password To Login



The screenshot shows the login page of the Plasma Donor App. The browser address bar displays '127.0.0.1:5000'. The page has a blue header with 'Plasma Donor App' on the left and 'Home' and 'Register' links on the right. The main content area features a dark blue background with a large yellow drop shape on the right. On the left, there is a white ECG line and the text 'EVERY DROP COUNTS' in white, followed by 'NOW FOR INDIA' in orange. The India Cares logo is in the top left. The login form is positioned inside the yellow drop, containing a text input for the name 'saravanan', a password input with masked characters, and a blue 'Login' button.

Registration Page:

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



The screenshot shows the registration page of the Plasma Donor App. The browser address bar displays '127.0.0.1:5000/registration'. The page has a blue header with 'Plasma Donor App' on the left and a 'Home' link on the right. The background is red with a pattern of red blood cells. The registration form is centered and includes the following fields: 'Enter Your Name', 'Enter Email', 'Enter 10-digit mobile number', 'Enter Your City Name', 'Select COVID infection status' (a dropdown menu), 'Choose your blood group' (a dropdown menu), and 'Enter Password'. A blue 'Register' button is located at the bottom of the form.

```
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)
```