Create IBM DB2 and Connect With Python

Login.html:

```
<!DOCTYPE html>
<html >
<!--From https://codepen.io/frytyler/pen/EGdtg-->
<head>
  <meta charset="UTF-8">
  <title>IBM Donor App</title>
        <link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>
        <link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>
        <link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet' type='text/css'>
        <link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300' rel='stylesheet' type='text/css'>
        <link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">


<style>
.login{
top: 20%;
}
</style>
</head>

<body>
<div class="header">
<div>Plasma Donor App</div>
        <ul>


                <li><a href="/registration">Register</a></li>
                <li><a class="active" href="/login">Home</a></li>
        </ul>
</div>
```

```
 <div class="login" >
                          <div>
                          </div>


    <!-- Main Input For Receiving Query to our ML -->
   <form action="{{ url_for('loginpage')}}"method="post">
             <input type="text" name="username" placeholder="Enter UserName" required="required"
style="color:black" />
       <input type="password" name="password" placeholder="Enter Password" required="required"
style="color:black" />
       <button type="submit" class="btn btn-primary btn-block btn-large">Login</button>


   </form>
<br><br>
<div style="color:black">
 {{ msg }}</div>
 </div>


</body>
</html>
```

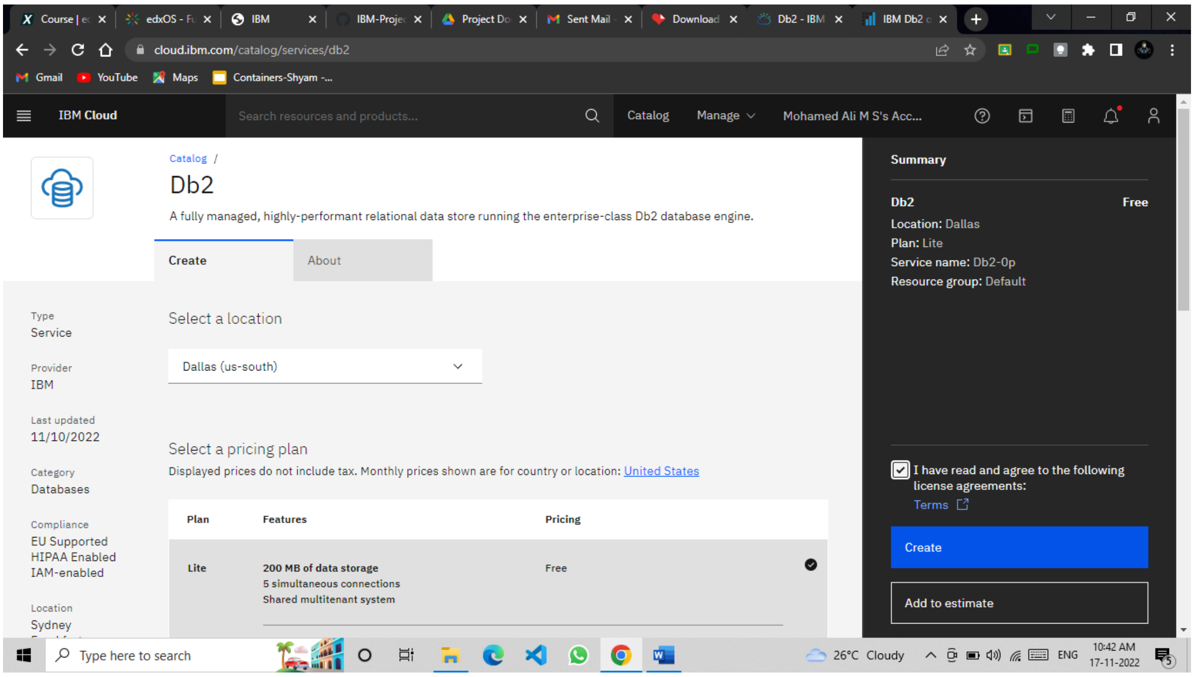
App.py

```
from distutils.log import debug
# from sendgridmail import sendmail
from flask import Flask, render_template, request, redirect, url_for, session
from flask_mail import Mail, Message
import re
import os
import ibm_db
from dotenv import load_dotenv


load_dotenv()



app = Flask(__name__)
```

```
app.secret_key = 'a'

print("Try to connect to Db2")

conn=ibm_db.connect("DATABASE=bludb;HOSTNAME=b1bc1829-6f45-4cd4-bef4-
10cf081900bf.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=32304;UID=ljh94224;SECURITY=S
SL;SSLServerCertificate=DigiCertGlobalRootCA.crt;PWD=xyRGrmo8S5w73Q20", ",")

print("Connected Successfully")


app.config['MAIL_SERVER']='smtp.gmail.com'

app.config['MAIL_PORT'] = 465

app.config['MAIL_USERNAME'] = 'cse19.mohamedmahroof@petengg.ac.in'

app.config['MAIL_PASSWORD'] = 'MOHAMEDMAHROOF'

app.config['MAIL_USE_TLS'] = False

app.config['MAIL_USE_SSL'] = True

mail = Mail(app)


@app.route('/')

@app.route('/login')

def login():


    return render_template('login.html')




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

def loginpage():

    global userid

    msg = "


    if request.method == 'POST' :

        username = request.form['username']

        password = request.form['password']

        sql = "SELECT * FROM donors 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 !'
            index(account['EMAIL'],'Plasma donor App login','You are successfully logged in!')
            return redirect(url_for('dash'))
        else:
            msg = 'Incorrect username / password !'
    return render_template('login.html', msg = msg)

@app.route('/registration')
def home():
    return render_template('register.html')

@app.route('/register',methods=['GET', 'POST'])
def register():
    msg = ''
    if request.method == 'POST' :
        username = request.form['username']
        email = request.form['email']
        password = request.form['password']
        phone = request.form['phone']
        city = request.form['city']
        infect = request.form['infect']
        blood = request.form['blood']
        sql = "SELECT * FROM donors 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("ac",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  donors VALUES (?, ?, ?, ?, ?, ?, ?)"
            prep_stmt = ibm_db.prepare(conn, insert_sql)
            ibm_db.bind_param(prep_stmt, 1, username)
            ibm_db.bind_param(prep_stmt, 2, password)
            ibm_db.bind_param(prep_stmt, 3, email)
            ibm_db.bind_param(prep_stmt, 4, phone)
            ibm_db.bind_param(prep_stmt, 5, city)
            ibm_db.bind_param(prep_stmt, 6, infect)
            ibm_db.bind_param(prep_stmt, 7, blood)

            ibm_db.execute(prep_stmt)
            msg = 'You have successfully registered, !'
            index(email,'Plasma donor App Registration','You are successfully Registered {}!'.format(username))

    elif request.method == 'POST':
        msg = 'Please fill out the form !'
    return render_template('register.html', msg = msg)


@app.route('/dashboard')
def dash():
    if session['loggedin'] == True:
        sql = "SELECT COUNT(*), (SELECT COUNT(*) FROM DONORS WHERE blood= 'O Positive'),
(SELECT COUNT(*) FROM DONORS WHERE blood='A Positive'), (SELECT COUNT(*) FROM
DONORS WHERE blood='B Positive'), (SELECT COUNT(*) FROM DONORS WHERE blood='AB
Positive'), (SELECT COUNT(*) FROM DONORS WHERE blood='O Negative'), (SELECT COUNT(*)
FROM DONORS WHERE blood='A Negative'), (SELECT COUNT(*) FROM DONORS WHERE blood='B
Negative'), (SELECT COUNT(*) FROM DONORS WHERE blood='AB Negative') FROM donors"
```

```
        stmt = ibm_db.prepare(conn, sql)

        ibm_db.execute(stmt)

        account = ibm_db.fetch_assoc(stmt)

        print(account)

        return render_template('dashboard.html',b=account)

    else:

        msg = 'Please login!'

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


@app.route('/requester')

def requester():

    if session['loggedin'] == True:

        return render_template('request.html')

    else:

        msg = 'Please login!'

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


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

def requested():

    bloodgrp = request.form['bloodgrp']

    address = request.form['address']

    name=  request.form['name']

    email=  request.form['email']

    phone= request.form['phone']

    insert_sql = "INSERT INTO  requested VALUES (?, ?, ?, ?, ?)"

    prep_stmt = ibm_db.prepare(conn, insert_sql)

    ibm_db.bind_param(prep_stmt, 1, bloodgrp)

    ibm_db.bind_param(prep_stmt, 2, address)

    ibm_db.bind_param(prep_stmt, 3, name)

    ibm_db.bind_param(prep_stmt, 4, email)

    ibm_db.bind_param(prep_stmt, 5, phone)

    ibm_db.execute(prep_stmt)

    index(email,'Plasma donor App plasma request','Your request for plasma is recieved.')

    return render_template('request.html', pred="Your request is sent to the concerned people.")
```

```
def index(usermail,subject,content):

   msg = Message(subject, sender = 'cse19.mohamedmahroof@petengg.ac.in', recipients = [usermail])
   msg.body = format(content)
   mail.send(msg)
   return "Sent"

@app.route('/logout')
def logout():
   session.pop('loggedin', None)
   session.pop('id', None)
   session.pop('username', None)
   return render_template('login.html')

if __name__ == '__main__':
   app.run(host='0.0.0.0',debug='TRUE')
```



Catalog /

Db2

A fully managed, highly-performant relational data store running the enterprise-class Db2 database engine.

Type
Service

Provider
IBM

Last updated
11/10/2022

Category
Databases

Compliance
EU Supported
HIPAA Enabled
IAM-enabled

Location
Sydney

Select a location

Dallas (us-south)

Select a pricing plan

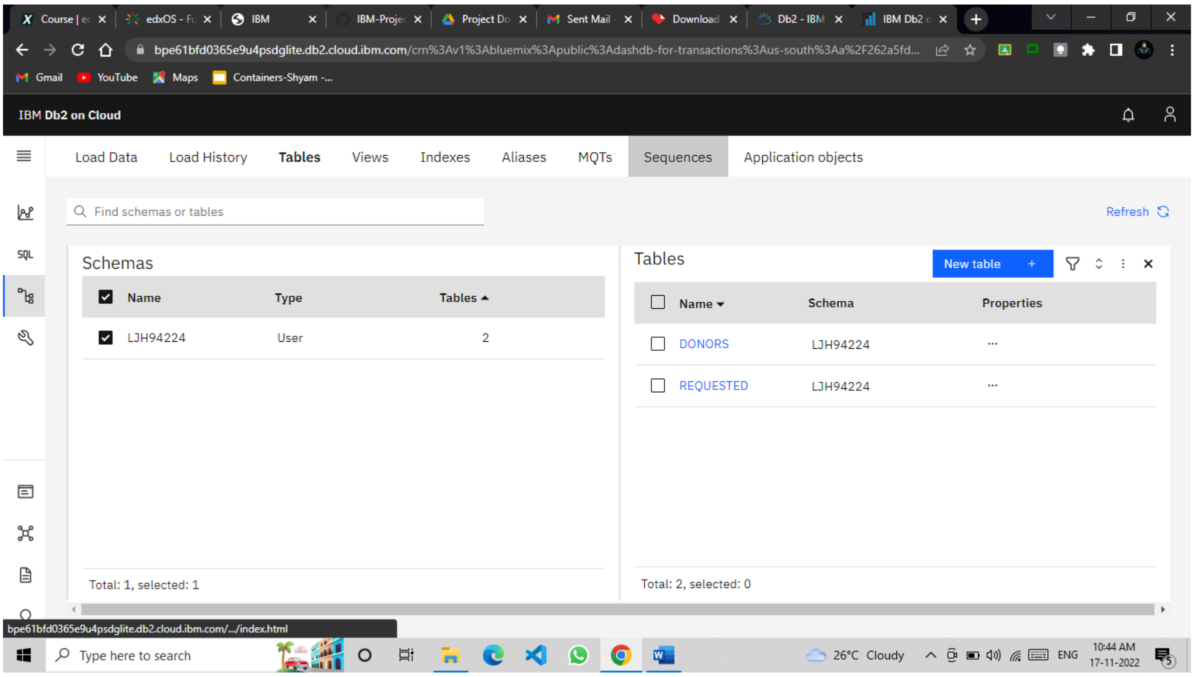
Displayed prices do not include tax. Monthly prices shown are for country or location: United States

Plan	Features	Pricing	
Lite	200 MB of data storage 5 simultaneous connections Shared multitenant system	Free	●

Summary

Db2 Free
Location: Dallas
Plan: Lite
Service name: Db2-0p
Resource group: Default

☑ I have read and agree to the following license agreements:
Terms



Schemas

☑	Name	Type	Tables ▲
☑	LJH94224	User	2

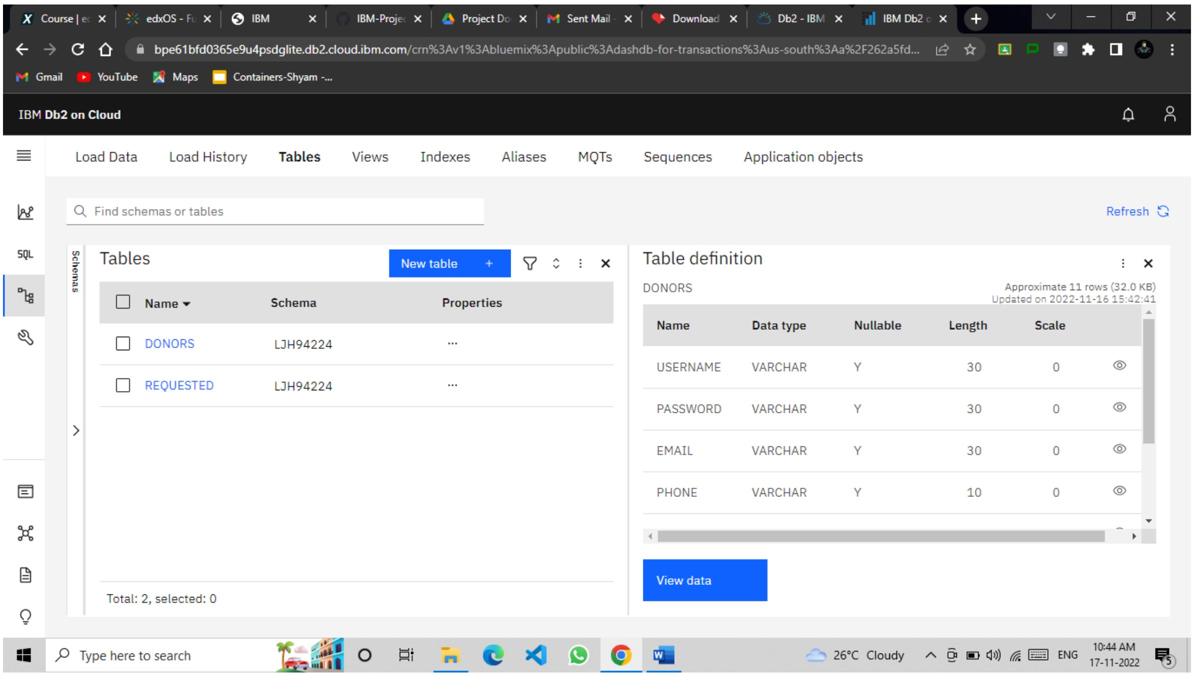
Total: 1, selected: 1

Tables

□	Name ▾	Schema	Properties
□	DONORS	LJH94224	···
□	REQUESTED	LJH94224	···

Total: 2, selected: 0

bpe61bfd0365e9u4psdglite.db2.cloud.ibm.com/.../index.html



Tables

□	Name ▾	Schema	Properties
□	DONORS	LJH94224	···
□	REQUESTED	LJH94224	···

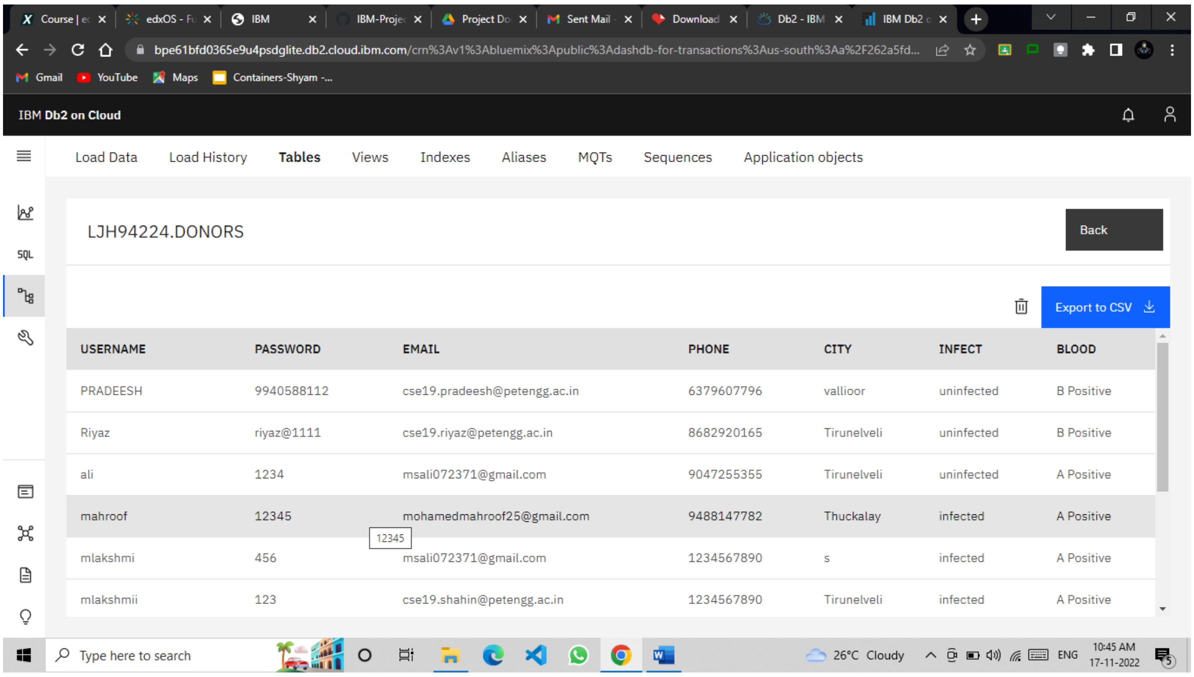
Total: 2, selected: 0

Table definition

DONORS

Approximate 11 rows (32.0 KB)
Updated on 2022-11-16 15:42:41

Name	Data type	Nullable	Length	Scale
USERNAME	VARCHAR	Y	30	0
PASSWORD	VARCHAR	Y	30	0
EMAIL	VARCHAR	Y	30	0
PHONE	VARCHAR	Y	10	0



LJH94224.DONORS

USERNAME	PASSWORD	EMAIL	PHONE	CITY	INFECT	BLOOD
PRADEESH	9940588112	cse19.pradeesh@petengg.ac.in	6379607796	vallioor	uninfected	B Positive
Riyaz	riyaz@1111	cse19.riyaz@petengg.ac.in	8682920165	Tirunelveli	uninfected	B Positive
ali	1234	msali072371@gmail.com	9047255355	Tirunelveli	uninfected	A Positive
mahroof	12345	mohamedmahroof25@gmail.com	9488147782	Thuckalay	infected	A Positive
mlakshmi	456	msali072371@gmail.com	1234567890	s	infected	A Positive
mlakshmii	123	cse19.shahin@petengg.ac.in	1234567890	Tirunelveli	infected	A Positive