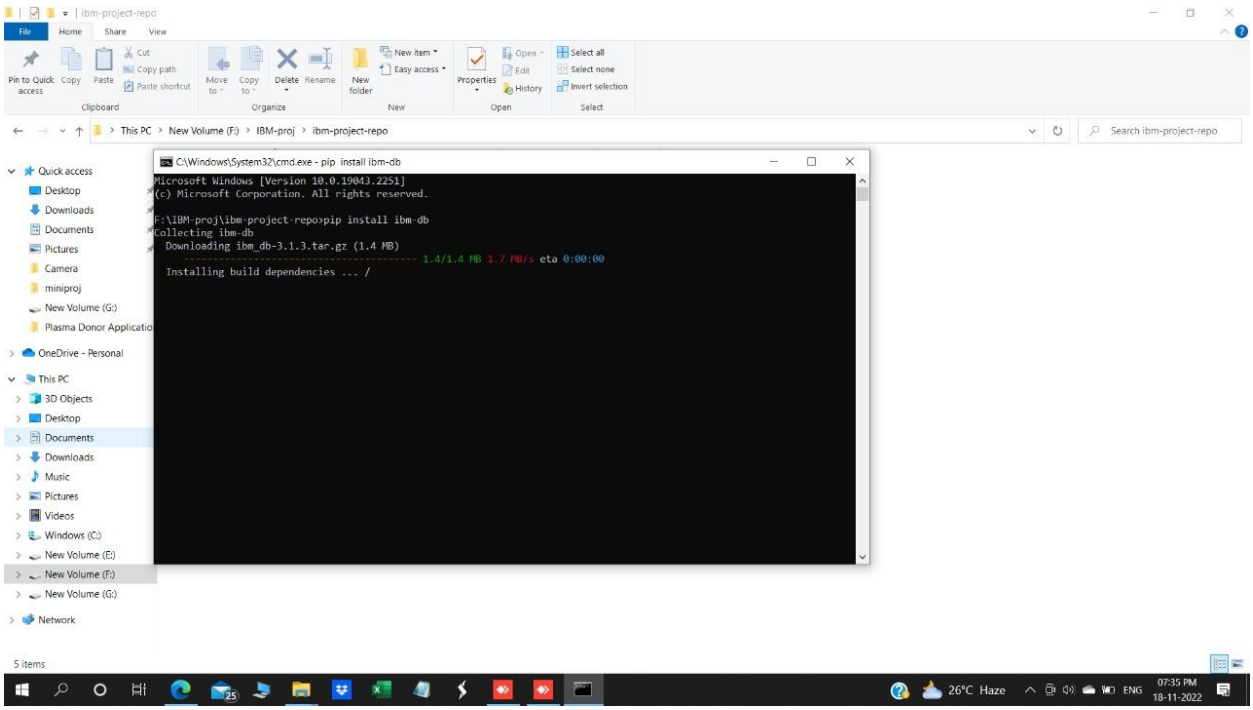
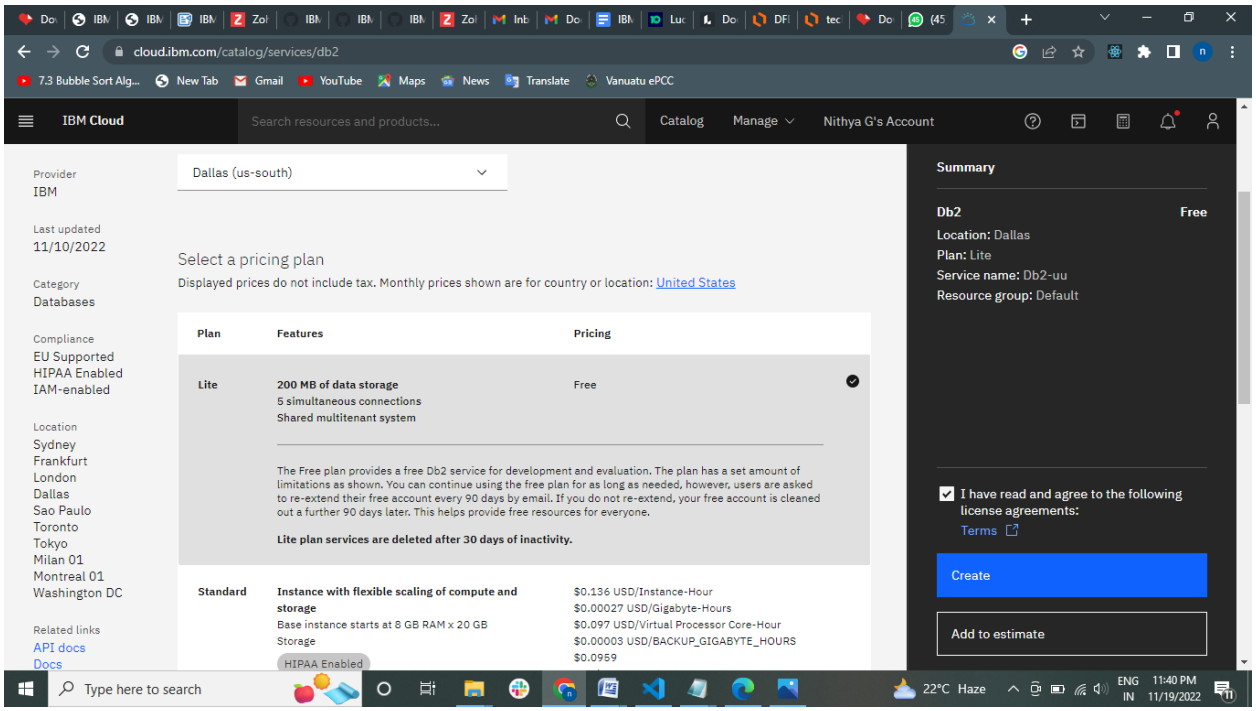
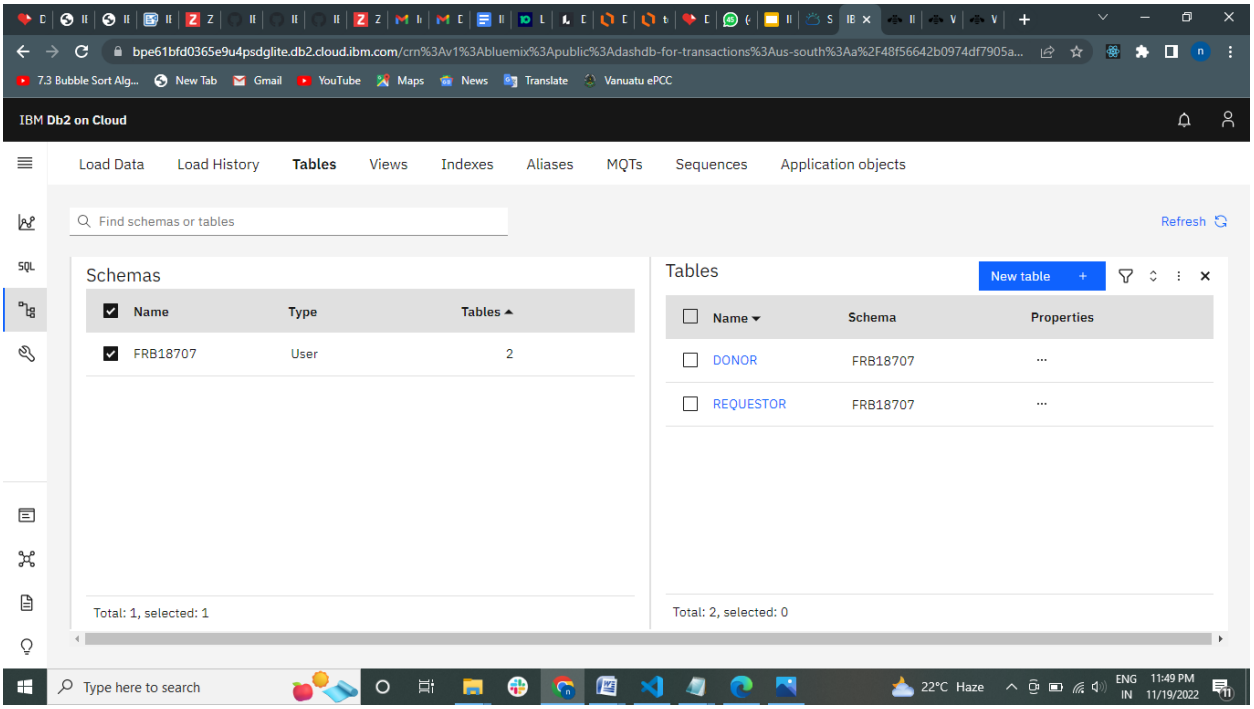
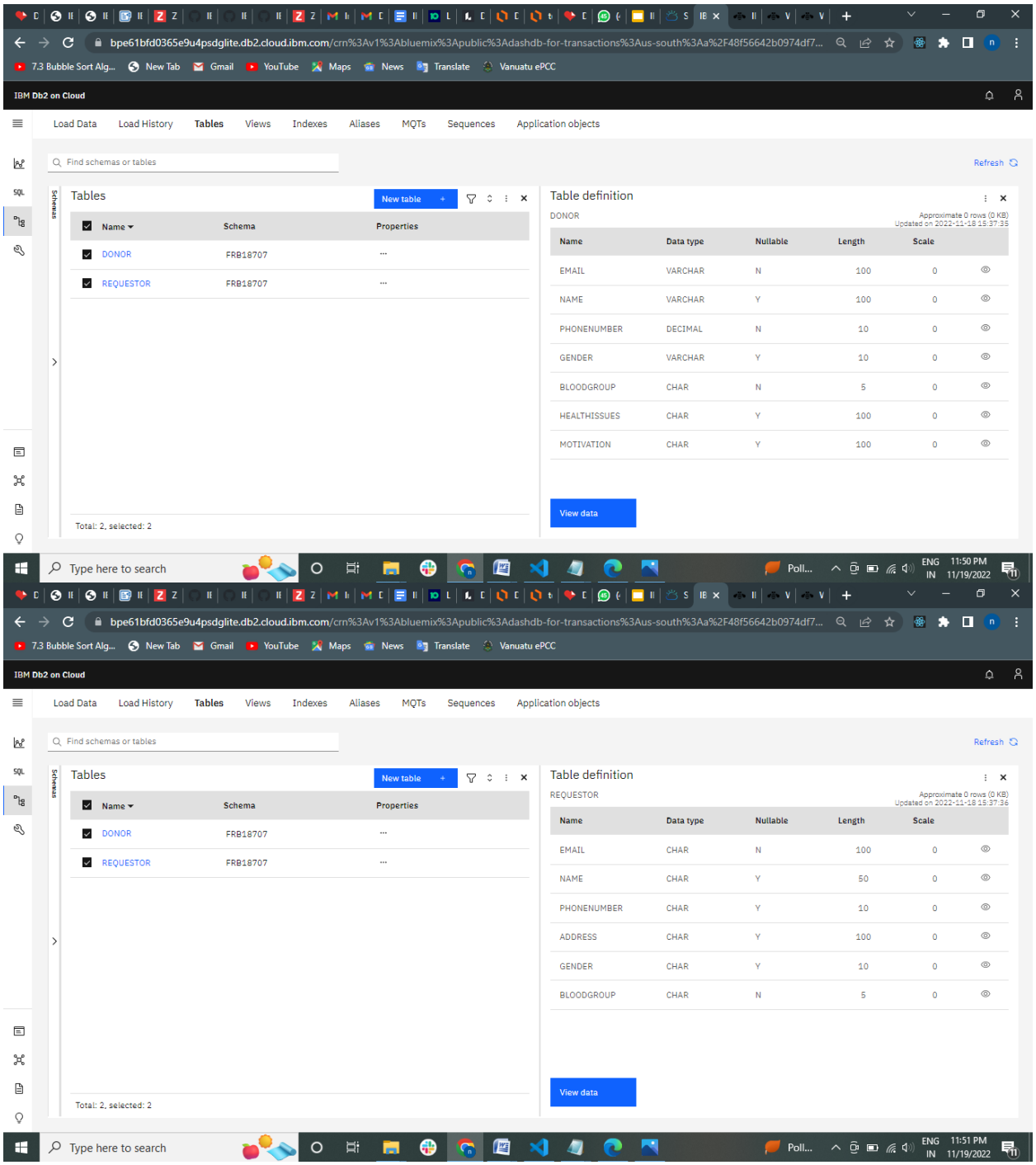
Date	12 Nov 2022
Team ID	PNT2022TMID20051
Project Name	Plasma Donor Application

Create IBM DB2 and connect with Python




cloud.ibm.com/catalog/services/db2
IBM Cloud
Search resources and products...
Catalog
Manage
Nithya G's Account
Provider
IBM
Last updated
11/10/2022
Category
Databases
Compliance
EU Supported
HIPAA Enabled
IAM-enabled
Location
Sydney
Frankfurt
London
Dallas
Sao Paulo
Toronto
Tokyo
Milan 01
Montreal 01
Washington DC
Related links
API docs
Docs
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
The Free plan provides a free Db2 service for development and evaluation. The plan has a set amount of limitations as shown. You can continue using the free plan for as long as needed, however, users are asked to re-extend their free account every 90 days by email. If you do not re-extend, your free account is cleaned out a further 90 days later. This helps provide free resources for everyone.
Lite plan services are deleted after 30 days of inactivity.
Standard
Instance with flexible scaling of compute and storage
Base instance starts at 8 GB RAM x 20 GB
Storage
HIPAA Enabled
$0.136 USD/Instance-Hour
$0.00027 USD/Gigabyte-Hours
$0.097 USD/Virtual Processor Core-Hour
$0.00003 USD/BACKUP_GIGABYTE_HOURS
$0.0959
Summary
Db2
Free
Location: Dallas
Plan: Lite
Service name: Db2-uu
Resource group: Default
I have read and agree to the following license agreements:
Terms
Create
Add to estimate
Type here to search
22°C Haze
ENG IN
11:40 PM 11/19/2022


bpe61bfd0365e9u4psdglite.db2.cloud.ibm.com/crn%3Av1%3Abluemix%3Apublic%3Adashdb-for-transactions%3Aus-south%3Aa%2F48f56642b0974df7905a...
IBM Db2 on Cloud
Load Data
Load History
Tables
Views
Indexes
Aliases
MQTs
Sequences
Application objects
Find schemas or tables
Refresh
Schemas
Name
Type
Tables
FRB18707
User
2
Total: 1, selected: 1
Tables
New table
Name
Schema
Properties
DONOR
FRB18707
REQUESTOR
FRB18707
Total: 2, selected: 0
Type here to search
22°C Haze
ENG IN
11:49 PM 11/19/2022




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

```
load_dotenv()


app = Flask(__name__)

app.secret_key = 'a'

conn=ibm_db.connect(os.getenv('DB_KEY'),"","")

@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 !'
            sendmail(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' :
        name = request.form['name']
        email = request.form['email']
        phone = request.form['phone']
        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(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]+', name):
            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, name)
            ibm_db.bind_param(prep_stmt, 2, email)
            ibm_db.bind_param(prep_stmt, 6, blood)
            ibm_db.bind_param(prep_stmt, 7, phone)

            ibm_db.execute(prep_stmt)
            msg = 'You have successfully registered !'
            sendmail(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)
    sendmail(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.")


@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')
```