

```

from flask import Flask, render_template, request,
session
import ibm_db
import re
import random
import yagmail
from time import gmtime, strftime

app=Flask(__name__)
app.secret_key = "a"
pass1='mlxpbmhzvrrkfzbq'

conn=ibm_db.connect("DATABASE=bludb;HOSTNAME=1bbf73c5-
d84a-4bb0-85b9-
ab1a4348f4a4.c3n41cmd0nqnrk39u98g.databases.appdomain.cl
oud;PORT=32286;SECURITY=SSL;SSLServerCERTIFICATE=DigiCer
tGlobalRootCA.crt;UID=ygb78843;PWD=kMybE3BM4ArGXlWQ;", ' '
, '')

@app.route('/')

def home():

    return render_template('homepage.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']
        uid=random.randint(0, 100)

        sql="SELECT * FROM credentials_1 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'[A-Za-z0-9]+', username):
        msg = 'name must contain only characters and
numbers !'
    else:

        #IBM
        insert_sql="INSERT INTO credentials_1
VALUES(?,?,?,?)"
        prep_stmt=ibm_db.prepare(conn,insert_sql)
        ibm_db.bind_param(prepare_stmt,1,uid)
        ibm_db.bind_param(prepare_stmt,2,username)
        ibm_db.bind_param(prepare_stmt,3,email)
        ibm_db.bind_param(prepare_stmt,4,password)
        ibm_db.execute(prepare_stmt)
        msg = 'You have successfully registered !'
        user = 'covidsfapp@gmail.com'
        app_password = pass1 # a token for gmail
        to = email
        subject = 'Successful Registration'
        content = ['Registered on ',strftime("At %Y-
%m-%d \n On %H:%M:%S", gmtime()))+" \n Login for further
usage" ]

        with yagmail.SMTP(user, app_password) as
yag:
            yag.send(to, subject, content)
            print('Sent email successfully')
            #works fine

    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 = ''
    user= ''
    #count updation

```

```

global activecases
global newcases
global deaths
global recovered

sql="SELECT * FROM updates_1 WHERE id = 1"
stmt=ibm_db.prepare(conn,sql)
ibm_db.execute(stmt)
update=ibm_db.fetch_assoc(stmt)
print(update)
activecases=ibm_db.result(stmt,0)
newcases=ibm_db.result(stmt,0)
deaths=ibm_db.result(stmt,0)
recovered=ibm_db.result(stmt,0)

if request.method == 'POST':
    username = request.form['username']
    password = request.form['password']

    sql="SELECT * FROM credentials_1 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'] =ibm_db.result(stmt,0)
        session['username'] = ibm_db.result(stmt,1)
        email=ibm_db.result(stmt,2)
        print(username)
        if re.match('admin1309',username):
            msg='welocome admin'
            return
    render_template('admin.html',msg=msg,activecases=activec
ases,newcases=newcases,deaths=deaths,recovered=recovered
,user=username)
    else:

```

```

        msg = 'Logged in successfully !'
        user = 'covidsfapp@gmail.com'
        app_password = pass1 # a token for gmail
        to = email
        subject = 'Logged in Successfully'
        content = ['Hey user '+username+' \n
Logged In ',strftime("At %Y-%m-%d \n On %H:%M:%S",
gmtime()))+"We are happy to serve you \n If this isn't
you please let us know at covidsafe@gmail.com" ]

        with yagmail.SMTP(user, app_password) as
yag:
            yag.send(to, subject, content)
            print('Sent email successfully')
            #works fine

        return render_template('dashboard.html',
msg=msg,activecases=activecases,newcases=newcases,deaths
=deaths,recovered=recovered,user=username)
    else:
        msg = 'Incorrect username / password !'
        return render_template('login.html',msg=msg)
@app.route('/dashboard_admin')
def dashboard_admin():
    global activecases
    global newcases
    global deaths
    global recovered

    sql="SELECT * FROM updates_1 WHERE id = 1"
    stmt=ibm_db.prepare(conn,sql)
    ibm_db.execute(stmt)
    update=ibm_db.fetch_assoc(stmt)
    print(update)
    activecases = ibm_db.result(stmt,1)
    newcases = ibm_db.result(stmt,2)
    deaths = ibm_db.result(stmt,3)
    recovered =ibm_db.result(stmt,4)
    username= session["username"]

```

```

        return
render_template('admin.html',user=username,activecases=activecases,newcases=newcases,deaths=deaths,recovered=recovered)

@app.route('/dashboard')
def dashboard():
    global activecases
    global newcases
    global deaths
    global recovered

    sql="SELECT * FROM updates_1 WHERE id = 1"
    stmt=ibm_db.prepare(conn,sql)
    ibm_db.execute(stmt)
    update=ibm_db.fetch_assoc(stmt)
    print(update)
    activecases = ibm_db.result(stmt,1)
    newcases = ibm_db.result(stmt,2)
    deaths = ibm_db.result(stmt,3)
    recovered =ibm_db.result(stmt,4)
    username= session["username"]
    return
render_template('dashboard.html',user=username,activecases=activecases,newcases=newcases,deaths=deaths,recovered=recovered)
@app.route('/verify',methods =['GET', 'POST'])
def verify():
    msg = ''
    username = session["username"]
    if request.method == 'POST':
        state = request.form['state']
        district = request.form['district']
        area = request.form['area']

        sql="SELECT * FROM zones_1 WHERE district = ?
AND area = ?"
        stmt=ibm_db.prepare(conn,sql)
        ibm_db.bind_param(stmt,1,district)
        ibm_db.bind_param(stmt,2,area)

```

```

        ibm_db.execute(stmt)
        zone=ibm_db.fetch_assoc(stmt)

        print(zone)
        if zone:
            msg = 'YOU ARE CURRENTLY IN A CONTAINMENT
ZONE !'

            insert_sql="INSERT INTO visited_1 VALUES
(?,?,?,?)"
            prep_stmt=ibm_db.prepare(conn,insert_sql)
            ibm_db.bind_param(prepare_stmt,1,username)
            ibm_db.bind_param(prepare_stmt,2,state)
            ibm_db.bind_param(prepare_stmt,3,district)
            ibm_db.bind_param(prepare_stmt,4,area)
            ibm_db.execute(prepare_stmt)
        else:
            msg = 'YOU ARE CURRENTLY NOT IN A
CONTAINMENT ZONE!'
        elif request.method == 'POST':
            msg = 'Please enter a zone !'
        return
render_template('verify.html',user=username,msg=msg)
@app.route('/visitedzones')
def visitedzones():
    username = session["username"]

    sql='SELECT * FROM visited_1 WHERE username = ? '
    stmt=ibm_db.prepare(conn,sql)
    ibm_db.bind_param(stmt,1,username)
    ibm_db.execute(stmt)
    visited=ibm_db.fetch_assoc(stmt)
    print(visited)
    if visited:
        visitedzones=str()
        v=0
        while visited:
            v=v+1

```

```

        visitedzones+=" =>" +str(v)+"
"+ibm_db.result(stmt,1)+"_" +ibm_db.result(stmt,2)+"_" +ib
m_db.result(stmt,3)+' , '
        if v==3:
            break
    else:
        visitedzones="YOU HAVE NOT VISITED ANY
CONTAINMENT ZONE"
        return render_template('visitedzones.html',
user=username, visitedzones=visitedzones)

@app.route('/countupdate',methods =['GET', 'POST'])
def countupdate():
    username = session["username"]
    if request.method == 'POST':
        activecases = request.form['activecases']
        newcases = request.form['newcases']
        deaths = request.form['deaths']
        recovered = request.form['recovered']

        update_sql="UPDATE updates_1 SET activecases =
?, newcases = ?, deaths = ?, recovered = ? WHERE id = 1"
        prep_stmt=ibm_db.prepare(conn,update_sql)
        ibm_db.bind_param(prepare_stmt,1,activecases)
        ibm_db.bind_param(prepare_stmt,2,newcases)
        ibm_db.bind_param(prepare_stmt,3,deaths)
        ibm_db.bind_param(prepare_stmt,4,recovered)
        ibm_db.execute(prepare_stmt)
    return
render_template('countupdate.html',user=username)
@app.route('/zoneupdate',methods =['GET', 'POST'])
def zoneupdate():
    msg = ''
    username=session["username"]
    if request.method == 'POST':
        state = request.form['state']
        district = request.form['district']
        area = request.form['area']

```

```

        sql='SELECT * FROM zones_1 WHERE district = ?
AND area = ?'
        stmt=ibm_db.prepare(conn,sql)
        ibm_db.bind_param(stmt,1,district)
        ibm_db.bind_param(stmt,2,area)
        ibm_db.execute(stmt)
        zone=ibm_db.fetch_assoc(stmt)

        print(zone)
        if zone:
            msg = 'zone already exists !'
        else:

            insert_sql="INSERT INTO zones_1 VALUES
(?,?,?)"
            prep_stmt=ibm_db.prepare(conn,insert_sql)
            ibm_db.bind_param(prep_stmt,1,state)
            ibm_db.bind_param(prep_stmt,2,district)
            ibm_db.bind_param(prep_stmt,3,area)
            ibm_db.execute(prep_stmt)
            msg = 'zone has been added !'
        elif request.method == 'POST':
            msg = 'Please enter a zone !'
        return
render_template('zoneupdate.html',msg=msg,user=username)

@app.route('/zonedeleletion',methods =['GET', 'POST'])
def zonedeleletion():
    msg=''
    username = session["username"]
    if request.method == 'POST':
        state = request.form['state']
        district = request.form['district']
        area = request.form['area']

        sql='SELECT * FROM zones_1 WHERE district = ?
AND area = ?'
        stmt=ibm_db.prepare(conn,sql)
        ibm_db.bind_param(stmt,1,district)
        ibm_db.bind_param(stmt,2,area)

```



```

        ibm_db.execute(stmt)
        zone=ibm_db.fetch_assoc(stmt)
        print(zone)
        if not zone:
            msg = 'zone does not exist !'
        else:
            del_sql='DELETE FROM zones_1 WHERE district
= ? AND area = ?'
            stmt=ibm_db.prepare(conn,del_sql)
            ibm_db.bind_param(stmt,1,district)
            ibm_db.bind_param(stmt,2,area)
            ibm_db.execute(stmt)
        elif request.method == 'POST':
            msg = 'Please enter a zone !'
        return
render_template('zonedeleation.html',msg=msg,user=username
e )

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

if __name__ == '__main__':
    app.debug = True
    app.run(port = 8080)

```