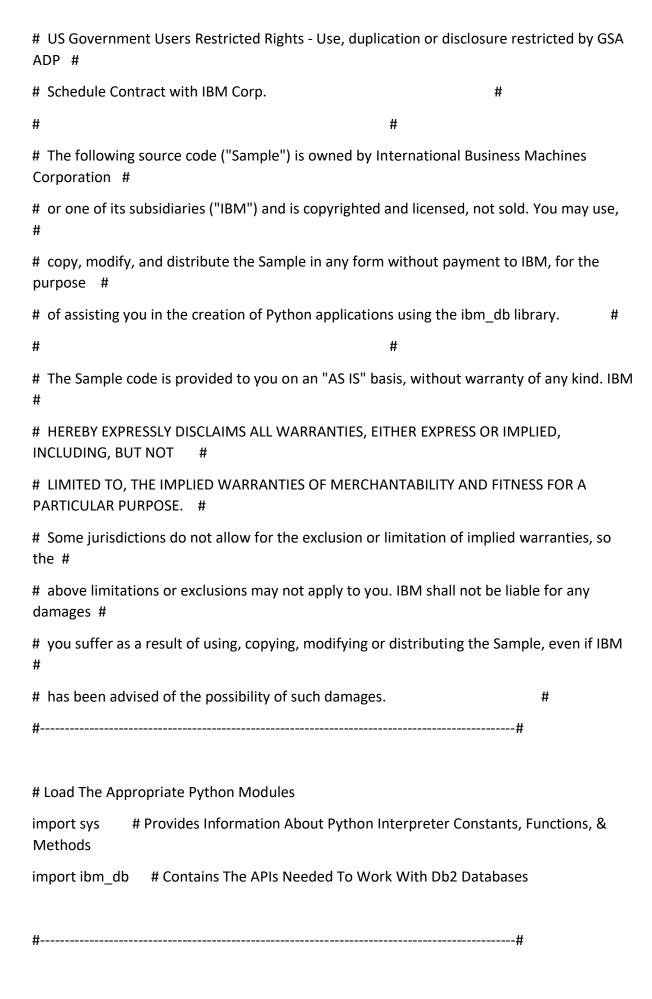
## IMPLEMENTING WEB APPLICATION

Project Date	September 2022		
Team ID	PNT2022TMID34297		
Project Name	Containment zone alerting application		

## **Create IBM DB2 And Connect With Python Code:**

#	! /usr/bin/python3				
#-			# #		
Ν	AME: ibm_db-connect_SERVER.py		#		
#		#			
# #	PURPOSE: This program is designed to illustrate h	ow to us	se the ibm_db.connect() API	to	
#	establish a connection to a Db2 server.		#		
#		#			
#	Additional APIs used:		#		
#	ibm_db.close()		#		
#		#			
# #	USAGE: Log in as a Db2 database instance user (f	or exam	nple, db2inst1) and issue the		
#	following command from a terminal window	:	#		
#		#			
#	./ibm_db-connect_SERVER.py		#		
#		#			
#-			#		
#	DISCLAIMER OF WARRANTIES AND LIMITATION OF LIABILITY #				
#		#			
# #	(C) COPYRIGHT International Business Machines Corp. 2018, 2019 All Rights Reserved				
#	Licensed Materials - Property of IBM		#		
#		#			



```
# Import The ipynb exit Class Definition, Attributes, And Methods That Have Been Defined
In The #
# File Named "ipynb_exit.py"; This Class Contains The Programming Logic Needed To Allow
# Functionality To Work Without Raising An Error Or Stopping The Kernel If The Application
                                                               # #-----
# Invoked In A Jupyter Notebook
  .-----#
from ipynb exit import exit
# Define And Initialize The Appropriate Variables hostName =
"197.126.80.22" # IP Address Of Remote Server
portNum = "50000"
                       # Port Number That Receives Db2 Connections On The Remote
Server userID = "db2inst2"
                             # The Instance User ID At The Remote Server passWord =
"ibmdb2"
              # The Password For The Instance User ID At The Remote Server
connectionID = None
# Display A Status Message Indicating An Attempt To Establish A Connection To A Db2
Server
# Is About To Be Made print("\nConnecting to the \"" + hostName
+ "\' server ... ", end="")
# Construct The String That Will Be Used To Establish A Db2 Server Connection connString
= "DRIVER={IBM DB2 ODBC DRIVER}"
connString += ";ATTACH=TRUE" # Attach To A Server; Not A Database
connString += ";DATABASE=" # Ignored When Connecting To A Server
connString += ";HOSTNAME=" + hostName  # Required To Connect To A Server
connString += ";PORT=" + portNum # Required To Connect To A Server
connString += ";PROTOCOL=TCPIP" # Required To Connect To A Server
connString += ";UID=" + userID connString += ";PWD=" + passWord
```

# Attempt To Establish A Connection To The Server Specified

```
try:
  connectionID = ibm_db.connect(connString, ", ") except
Exception:
  pass
# If A Db2 Server Connection Could Not Be Established, Display An Error Message And Exit
if connectionID is None:
  print("\nERROR: Unable to connect to the \"" + hostName + "\' server.")
print("Connection string used: " + connString + "\n")
  exit(-1)
# Otherwise, Complete The Status Message
else:
  print("Done!\n")
# Add Additional Db2 Server-Related Processing Here ...
# For Example, ibm db.createdb(), ibm db.createdbNX(), ibm db.recreatedb(),
ibm_db.dropdb()
# Attempt To Close The Db2 Server Connection That Was Just Opened if
not connectionID is None:
  print("Disconnecting from the \"" + hostName + "\' server ... ", end="")
  try:
    returnCode = ibm_db.close(connectionID)
except Exception:
    pass
  # If The Db2 Server Connection Was Not Closed, Display An Error Message And Exit
  if returnCode is False:
    print("\nERROR: Unable to disconnect from the " + hostName + " server.")
```

```
exit(-1)

# Otherwise, Complete The Status Message
else:
    print("Done!\n")

# Return Control To The Operating System
exit()
```