Python Script for Employee Separation Reason API

!pip install azure-keyvault

import pandas as pd

import pyodbc

import numpy as np

import json

from azure.identity import ClientSecretCredential

from azure.keyvault.secrets import SecretClient

from azure.identity import ClientSecretCredential

import time

from datetime import datetime, timedelta

KEYVAULT\_URI = "https://std-inc-hr-kv-01.vault.azure.net/"

tenant\_id = '1250f2eb-4784-4223-98dc-d6e33677888x'

client\_id = '969c6875-c148-4a09-9d94-66c8f265rTwx6'

client\_secret = '~PX8Q~YzQ\_WjJr1cmd8SH3WpChBAjwUbFGV56gs12P'

credential = ClientSecretCredential(tenant\_id=tenant\_id, client\_id=client\_id, client\_secret=client\_secret)

client = SecretClient(vault\_url=KEYVAULT\_URI, credential=credential)

DEMO\_DB\_PASSWORD=client.get\_secret("STD-INC-HR-SQLDB-01-SK-01").value

server = 'std-inc-hr-sqldbsrv-01.database.windows.net'

db1 = 'STD-INC-HR-SQLDB-01'

username = 'hrsqladmin'

password = DEMO\_DB\_PASSWORD

cnxn = pyodbc.connect('DRIVER={ODBC Driver 18 for SQL Server};SERVER='+server+';DATABASE='+db1+';ENCRYPT=no;UID='+username+';PWD='+ password)

cur = cnxn.cursor()

today = datetime.today()

yesterday = today - timedelta(days=1)

yesterday=yesterday.strftime('%d-%m-%Y')

yesterday

import requests

from datetime import date, timedelta

url = "https://myjsw.darwinbox.in/UpdateEmployeeDetails/getformsdata"

# Calculate yesterday's date

yesterday = date.today() - timedelta(days=1)

yesterday\_str = yesterday.strftime("%d-%m-%Y")

# Calculate today's date

today = date.today()

today\_str = today.strftime("%d-%m-%Y")

payload = {

    "api\_key": "19f5364c35950c09f9616a402b40a75e6aafdbde56c1f83f7566fc821213190570c9580353816edebb840e7ca1aa35da502bc5e9adba074efa8c73edFGV234ew",

    "form\_id": "a63638893228f3",

    "type": "separation\_approval",

    "form\_type": "separation\_approval-form",

    "from": yesterday\_str,

    "to": today\_str

}

headers = {

    'Content-Type': 'application/json',

    'Authorization': 'Basic anN3X2dyY19mb3Jtc191c2VyOnhGTVU0SkAqbHIxaWtXUVgpVGdMbV4yQ3E='

}

response = requests.post(url, headers=headers, json=payload)

print(response.text)

data = response.json()

# Extract column names and data from the JSON response

cols = data['cols']

output = data['output']

# Create a dataframe from the data

df = pd.DataFrame(output, columns=cols)

df

df = df.astype(object)

#for df1

col1 = ", ".join([str(i.replace('/','\_').replace('.','').replace(' ','\_')) for i in df.columns.tolist()])

col1

cur.execute("Truncate TABLE Stagging\_Employee\_Separation;")

cnxn.commit()

#insertion for df1(Active)

for i,row in df.iterrows():

    sql = "INSERT INTO Stagging\_Employee\_Separation (" +col1 + ") VALUES (" + "?,"\*(len(row)-1) + "?)"

    cur.execute(sql, tuple(row))

cnxn.commit()

cur.execute("Update\_Production\_Employee\_Separation;")

cnxn.commit()