**Extranet Website Code:**

from re import sub

import ssl

import requests

import pandas as pd

from bs4 import BeautifulSoup

import pprint

import xml.etree.ElementTree as ET

import certifi

import urllib

g = []

wa1='https://extranet.edqm.eu/4DLink1/4DCGI/Query\_CEP?vSelectName=1&Case\_TSE=none&vContains=1&vContainsDate=1&vtsubName=Tiotropium+bromide&vtsubDateBegin=&vtsubDateBtwBegin=&vtsubDateBtwEnd=&SWTP=1&OK=Search'

try:

\_create\_unverified\_https\_context = ssl.\_create\_unverified\_context

except AttributeError:

pass

else:

ssl.\_create\_default\_https\_context = \_create\_unverified\_https\_context

for i in range(1, 2):

wa = wa1 + str(i)

#print(wa)

x = urllib.request.urlopen(wa).read()

response = requests.get(wa,verify=False)

rt = response.text

soup = BeautifulSoup(rt, "html.parser")

tables = soup.find\_all('table')[2].find\_all('tr')

#print(tables)

for table in tables:

h=[]

for row in table.find\_all('td'):

try:

dat = row.text

#print(dat)

except:

dat = ""

h.append(dat)

g.append(h[0:8])

df = pd.DataFrame(g, columns=['Substance Number', 'Substance ', 'Certificate Holder', 'Certificate Number','Issue Date','Status','End date','Type'])

df1 = df.transpose()

df.to\_excel('C:/Python/Python/excel/data/Tiotropium bromide.xlsx')

**ClinicalTrial Website Code:**

from re import sub

import ssl

import requests

import pandas as pd

from bs4 import BeautifulSoup

import pprint

import xml.etree.ElementTree as ET

import certifi

import urllib

g = []

#wa1 = 'https://www.clinicaltrialsregister.eu/ctr-search/search?query=&page='

wa1 = 'https://www.clinicaltrialsregister.eu/ctr-search/search?query=&phase=phase-two&page='

try:

\_create\_unverified\_https\_context = ssl.\_create\_unverified\_context

except AttributeError:

# Legacy Python that doesn't verify HTTPS certificates by default

pass

else:

# Handle target environment that doesn't support HTTPS verification

ssl.\_create\_default\_https\_context = \_create\_unverified\_https\_context

#ssl.SSLContext.verify\_mode = property(lambda self: ssl.CERT\_NONE, lambda self, newval: None)

for i in range(1,951):

wa = wa1 + str(i)

#print(wa)

x = urllib.request.urlopen(wa).read()

# print(x)

response = requests.get(wa,verify=False)

rt = response.text

soup = BeautifulSoup(rt, "html.parser")

tables = soup.find\_all('table', class\_="result")

#print(tables)

for table in tables:

h=[]

for row in table.find\_all('tr'):

for column in row.find\_all('td'):

try:

dat = column.text.strip().split(':')[1]

except:

dat = ""

h.append(dat)

g.append(h[0:6])

#print(g)

df = pd.DataFrame(g, columns=['EudraCT Number', 'Protocol Number', 'Start Date', 'Sponsor Name','Title','Medical Condition'])

df1 = df.transpose()

# print("Print to excel")

df.to\_excel('clinicaltrials.xlsx')

**CTRI Website**

from re import sub

import requests

import pandas as pd

from bs4 import BeautifulSoup

import pprint

import xml.etree.ElementTree as ET

wa = 'http://www.ctri.nic.in/Clinicaltrials/pmaindet2.php?trialid=50774'

#print(wa)

class HTMLTableParser:

def parse\_url(self, url):

response = requests.get(url)

soup = BeautifulSoup(response.text, 'lxml')

return [(table['id'],self.parse\_html\_table(table))\

for table in soup.find\_all('table')]

def parse\_subtable\_h(self, table):

sub\_dir = []

first\_row = 1

for row in table.find\_all('tr'):

ostr = ""

for column in row.find\_all('td'):

sub\_dir.append(column.text)

return sub\_dir

def parse\_html\_table(self, table):

df = {}

row\_index = 0

edit\_vals = {10:5, 20:3}

pharma\_trial = 0

for row in table.find\_all('tr', recursive=False):

columns = row.find\_all('td')

c0 = columns[0].text.replace('\n','').strip()

subtable\_found = 0

try:

subtable = columns[1].find\_all('table')[0]

subtable\_found = 1

#print('Subtable found', subtable)

except:

subtable\_found = 0

pass

if (subtable\_found == 1):

#print("Subtable found")

sub\_text = self.parse\_subtable\_h(subtable)

if row\_index in edit\_vals.keys():

c1 = sub\_text[edit\_vals[row\_index]]

elif row\_index == 14:

c1 = sub\_text[1]

df['Sponsor Address'] = sub\_text[3]

df['Sponsor Type'] = sub\_text[5]

#print(sub\_text[5][:10])

if sub\_text[5][:14] == "Pharmaceutical":

print("pharma trial")

pharma\_trial = 1

elif row\_index == 21:

if pharma\_trial == 1:

try:

cindx = sub\_text.index('Intervention\xa0')

c1 = sub\_text[cindx + 1]

except:

c1 = "Could not parse details"

else:

c1 = ""

else:

c1 = ", ".join(sub\_text)

else:

c1 = columns[1].text

if row\_index == 0:

c1 = c1[:20]

df[c0] = c1.strip('\n')

row\_index += 1

return df

with open("C:/Python/Python/ctri.mht") as f:

rt1 = f.read()

#print(rt)

print(type(rt1))

import re

g=[m.start() for m in re.finditer("value=3D", rt1)]

#print(g)

print(len(g))

wa1 = "http://www.ctri.nic.in/Clinicaltrials/pmaindet2.php?trialid="

df = pd.DataFrame()

for x in g:

trialID = rt1[x+9:x+14]

wa = wa1 + str(trialID)

print(wa)

ctri\_study = 0

try:

response = requests.get(wa)

rt = response.text

ctri\_study = 1

except:

print('no response')

if (ctri\_study):

soup = BeautifulSoup(rt, 'html.parser')

table = soup.find\_all('table')[2]

#print(table)

hp = HTMLTableParser()

df1 = hp.parse\_html\_table(table)

#print(df1)

df=pd.concat([df, pd.DataFrame([df1])])

df['link'] = wa

"""

try:

soup = BeautifulSoup(rt, 'html.parser')

table = soup.find\_all('table')[2]

print(table)

hp = HTMLTableParser()

df1 = hp.parse\_html\_table(table)

print(df1)

df=pd.concat([df, pd.DataFrame([df1])])

df['link'] = wa

except:

print("Could nto parse: " + wa)

pass

"""

print("Print to excel")

df.to\_excel('test.xlsx')