1. PROGRAM CODE (CREATED INDEPENDENTLY OR COLLABORATIVELY)

This file is without_scheduler.py which is used to run on local host to get instant result.

The code is as follows:

```
import pymongo
from bs4 import BeautifulSoup
import requests
from datetime import datetime
datetime_object = str(datetime.now()).replace(" ","").replace(".","").replace(":","")
dbname = 'covid-19-'+datetime_object
client =
pymongo.MongoClient("mongodb+srv://admin:admin@cluster0-kwhwi.azure.mongodb.net/test?
retryWrites=true&w=majority")

db = client.covid19

db.create_collection(dbname)

covid_col = db[dbname]

page = requests.get("https://www.worldometers.info/coronavirus/")


soup = BeautifulSoup(page.content,'html.parser')


table = soup.find_all('table')[0]

import pandas as pd

table_heads = table.find_all('th')
table_heads = [x.get_text() for x in table_heads]


table_rows = table.find_all('tr')

row_data = []
row_data_json = []
iterrows = iter(table_rows)
next(iterrows)
for row in iterrows:
```

```
    nl = []
    for data in row.find_all('td'):
        nl.append(data.get_text().strip())
    res = dict(zip(table_heads,nl))
    row_data.append(nl)
    row_data_json.append(res)

frame = pd.DataFrame(row_data,columns=table_heads)
frame.to_csv(dbname+'.csv')
frame.to_excel(dbname+'.xlsx')
frame.to_json(dbname+'.json')
print('created files')

covid_col.insert_many(row_data_json)
print(dbname + 'inserted into mongoDB')

    # for row in row_data_json:
    #     covid_col.insert_one(row)
    #     print('inserted')
msg = 'Task Completed at \t' + str(datetime.now()) + '\n' + 'CREATED ' + dbname+'.csv' + '\n' +
'CREATED ' + dbname+'.xlsx\n' + 'CREATED ' + dbname+'.json' + '\n' + 'INSERTED data into
MongoDB in collection : ' + dbname + '\n'
print(msg)
import slack
#     import nest_asyncio
#     nest_asyncio.apply()

channel = '#herokuapp'

client =
slack.WebClient(token='xoxp-1033025358789-1023046500033-1033332113781-61f1d9d8f6d40
33a3c32fd68457ed0b5')

response = client.chat_postMessage(
    channel=channel,
    text=msg)
assert response["ok"]

response = client.files_upload(
    channels=channel,
    file=dbname+'.csv',
    title=dbname+'.csv')
assert response["ok"]
```

```
response = client.files_upload(
   channels=channel,
   file=dbname+'.xlsx',
   title=dbname+'.xlsx')
assert response["ok"]


response = client.files_upload(
   channels=channel,
   file=dbname+'.json',
   title=dbname+'.json')
assert response["ok"]

ss = covid_col.find({}, {'Country,Other':1,'TotalCases':1,'_id':0})

countries =[]
totalcases = []
for v in ss:
   countries.append(v['Country,Other'])
   totalcases.append(int(v['TotalCases'].replace(',','')))

import matplotlib.pyplot as plt
fig = plt.figure()
ax = fig.add_axes([0,0,2,2])
plt.xticks(rotation=90)
plt.xlabel('Country',color='black', fontweight='bold',fontsize=18)
plt.ylabel('Total Cases of COVID-19',color='black', fontweight='bold',fontsize=18)
plt.title(str(datetime.now().strftime("%Y-%m-%d %H:%M")),color='black', fontweight='bold')
ax.bar(countries[:20],totalcases[:20])
ss = totalcases[:20]
for i, v in enumerate(ss):
   ax.text(i-.30,v/ss[i]+1200, str(v), color='white', fontweight='bold',fontsize=8)
plt.savefig(dbname+'.png', bbox_inches='tight')
plt.show()

response = client.files_upload(
   channels=channel,
   file=dbname+'.png',
   title=dbname+'.png')
assert response["ok"]
```

This files is heroku_scheduler
It is uploaded on cloud and is executed automatically after a time span of 6 hrs.

```
from apscheduler.schedulers.blocking import BlockingScheduler

sched = BlockingScheduler()


@sched.scheduled_job('cron', day_of_week='*', hour=3)
@sched.scheduled_job('cron', day_of_week='*', hour=9)
@sched.scheduled_job('cron', day_of_week='*', hour=15)
@sched.scheduled_job('cron', day_of_week='*', hour=21)
def scheduled_job():
    import pymongo

    from bs4 import BeautifulSoup
    import requests
    from datetime import datetime
    datetime_object = str(datetime.now()).replace(" ","").replace(".","").replace(":","")
    dbname = 'covid-19-'+datetime_object
    client =
pymongo.MongoClient("mongodb+srv://admin:admin@cluster0-kwhwi.azure.mongodb.net/test?
retryWrites=true&w=majority")

    db = client.covid19

    db.create_collection(dbname)

    covid_col = db[dbname]

    page = requests.get("https://www.worldometers.info/coronavirus/")


    soup = BeautifulSoup(page.content,'html.parser')


    table = soup.find_all('table')[0]

    import pandas as pd

    table_heads = table.find_all('th')
```

```
    table_heads = [x.get_text() for x in table_heads]


    table_rows = table.find_all('tr')

    row_data = []
    row_data_json = []
    iterrows = iter(table_rows)
    next(iterrows)
    for row in iterrows:
        nl = []
        for data in row.find_all('td'):
            nl.append(data.get_text().strip())
        res = dict(zip(table_heads,nl))
        row_data.append(nl)
        row_data_json.append(res)

    frame = pd.DataFrame(row_data,columns=table_heads)
    frame.to_csv(dbname+'.csv')
    frame.to_excel(dbname+'.xlsx')
    frame.to_json(dbname+'.json')
    print('created files')

    covid_col.insert_many(row_data_json)
    print(dbname + 'inserted into mongoDB')

    # for row in row_data_json:
    #     covid_col.insert_one(row)
    #     print('inserted')
    msg = 'Task Completed at \t' + str(datetime.now()) + '\n' + 'CREATED ' + dbname+'.csv' + '\n'
+ 'CREATED ' + dbname+'.xlsx\n' + 'CREATED ' + dbname+'.json' + '\n' + 'INSERTED data into
MongoDB in collection : ' + dbname + '\n'
    print(msg)

    import slack
#     import nest_asyncio
#     nest_asyncio.apply()

    channel = '#herokuapp'

    client =
slack.WebClient(token='xoxp-1033025358789-1023046500033-1033332113781-61f1d9d8f6d40
33a3c32fd68457ed0b5')
```

```
response = client.chat_postMessage(
    channel=channel,
    text=msg)
assert response["ok"]

response = client.files_upload(
    channels=channel,
    file=dbname+'.csv',
    title=dbname+'.csv')
assert response["ok"]


response = client.files_upload(
    channels=channel,
    file=dbname+'.xlsx',
    title=dbname+'.xlsx')
assert response["ok"]


response = client.files_upload(
    channels=channel,
    file=dbname+'.json',
    title=dbname+'.json')
assert response["ok"]

ss = covid_col.find({}, {'Country,Other':1,'TotalCases':1,'_id':0})

countries =[]
totalcases = []
for v in ss:
    countries.append(v['Country,Other'])
    totalcases.append(int(v['TotalCases'].replace(',','')))

import matplotlib.pyplot as plt
fig = plt.figure()
ax = fig.add_axes([0,0,2,2])
plt.xticks(rotation=90)
plt.xlabel('Country',color='black', fontweight='bold',fontsize=18)
plt.ylabel('Total Cases of COVID-19',color='black', fontweight='bold',fontsize=18)
plt.title(str(datetime.now().strftime("%Y-%m-%d %H:%M")),color='black', fontweight='bold')
ax.bar(countries[:20],totalcases[:20])
ss = totalcases[:20]
for i, v in enumerate(ss):
    ax.text(i-.30,v/ss[i]+1200, str(v), color='white', fontweight='bold',fontsize=8)
```

```
    plt.savefig(dbname+'.png', bbox_inches='tight')
    plt.show()

    response = client.files_upload(
        channels=channel,
        file=dbname+'.png',
        title=dbname+'.png')
    assert response["ok"]



sched.start()
```