from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.common.keys import Keys

import pandas as pd

import time

import datetime

import smtplib

from email.mime.multipart import MIMEMultipart

browser = webdriver.Chrome(executable\_path='/chromedriver')

#Setting ticket types paths

return\_ticket = "//label[@id='flight-type-roundtrip-label-hp-flight']"

one\_way\_ticket = "//label[@id='flight-type-one-way-label-hp-flight']"

multi\_ticket = "//label[@id='flight-type-multi-dest-label-hp-flight']"

def ticket\_chooser(ticket):

try:

ticket\_type = browser.find\_element\_by\_xpath(ticket)

ticket\_type.click()

except Exception as e:

Pass

def dep\_country\_chooser(dep\_country):

fly\_from = browser.find\_element\_by\_xpath("//input[@id='flight-origin-hp-flight']")

time.sleep(1)

fly\_from.clear()

time.sleep(1.5)

fly\_from.send\_keys(' ' + dep\_country)

time.sleep(1.5)

first\_item = browser.find\_element\_by\_xpath("//a[@id='aria-option-0']")

time.sleep(1.5)

first\_item.click()

def arrival\_country\_chooser(arrival\_country):

fly\_to = browser.find\_element\_by\_xpath("//input[@id='flight-destination-hp-flight']")

time.sleep(1)

fly\_to.clear()

time.sleep(1.5)

fly\_to.send\_keys(' ' + arrival\_country)

time.sleep(1.5)

first\_item = browser.find\_element\_by\_xpath("//a[@id='aria-option-0']")

time.sleep(1.5)

first\_item.click()

def dep\_date\_chooser(month, day, year):

dep\_date\_button = browser.find\_element\_by\_xpath("//input[@id='flight-departing-hp-flight']")

dep\_date\_button.clear()

dep\_date\_button.send\_keys(month + '/' + day + '/' + year)

def return\_date\_chooser(month, day, year):

return\_date\_button = browser.find\_element\_by\_xpath("//input[@id='flight-returning-hp-flight']")

for i in range(11):

return\_date\_button.send\_keys(Keys.BACKSPACE)

return\_date\_button.send\_keys(month + '/' + day + '/' + year)

def search():

search = browser.find\_element\_by\_xpath("//button[@class='btn-primary btn-action gcw-submit']")

search.click()

time.sleep(15)

print('Results ready!')

df = pd.DataFrame()

def compile\_data():

global df

global dep\_times\_list

global arr\_times\_list

global airlines\_list

global price\_list

global durations\_list

global stops\_list

global layovers\_list

#departure times

dep\_times = browser.find\_elements\_by\_xpath("//span[@data-test-id='departure-time']")

dep\_times\_list = [value.text for value in dep\_times]

#arrival times

arr\_times = browser.find\_elements\_by\_xpath("//span[@data-test-id='arrival-time']")

arr\_times\_list = [value.text for value in arr\_times]

#airline name

airlines = browser.find\_elements\_by\_xpath("//span[@data-test-id='airline-name']")

airlines\_list = [value.text for value in airlines]

#prices

prices = browser.find\_elements\_by\_xpath("//span[@data-test-id='listing-price-dollars']")

price\_list = [value.text.split('$')[1] for value in prices]

#durations

durations = browser.find\_elements\_by\_xpath("//span[@data-test-id='duration']")

durations\_list = [value.text for value in durations]

#stops

stops = browser.find\_elements\_by\_xpath("//span[@class='number-stops']")

stops\_list = [value.text for value in stops]

#layovers

layovers = browser.find\_elements\_by\_xpath("//span[@data-test-id='layover-airport-stops']")

layovers\_list = [value.text for value in layovers]

now = datetime.datetime.now()

current\_date = (str(now.year) + '-' + str(now.month) + '-' + str(now.day))

current\_time = (str(now.hour) + ':' + str(now.minute))

current\_price = 'price' + '(' + current\_date + '---' + current\_time + ')'

for i in range(len(dep\_times\_list)):

try:

df.loc[i, 'departure\_time'] = dep\_times\_list[i]

except Exception as e:

pass

try:

df.loc[i, 'arrival\_time'] = arr\_times\_list[i]

except Exception as e:

pass

try:

df.loc[i, 'airline'] = airlines\_list[i]

except Exception as e:

pass

try:

df.loc[i, 'duration'] = durations\_list[i]

except Exception as e:

pass

try:

df.loc[i, 'stops'] = stops\_list[i]

except Exception as e:

pass

try:

df.loc[i, 'layovers'] = layovers\_list[i]

except Exception as e:

pass

try:

df.loc[i, str(current\_price)] = price\_list[i]

except Exception as e:

pass

print('Excel Sheet Created!')

now = datetime.datetime.now()

current\_date = (str(now.year) + '-' + str(now.month) + '-' + str(now.day))

current\_time = (str(now.hour) + ':' + str(now.minute))

current\_price = 'price' + '(' + current\_date + '---' + current\_time + ')'

#email credentials

username = 'myemail@hotmail.com'

password = 'XXXXXXXXXXX'

def connect\_mail(username, password):

global server

server = smtplib.SMTP('smtp.outlook.com', 587)

server.ehlo()

server.starttls()

server.login(username, password)

#Create message template for email

def create\_msg():

global msg

msg = '\nCurrent Cheapest flight:\n\nDeparture time: {}\nArrival time: {}\nAirline: {}\nFlight duration: {}\nNo. of stops: {}\nPrice: {}\n'.format(cheapest\_dep\_time,

cheapest\_arrival\_time,

cheapest\_airline,

cheapest\_duration,

cheapest\_stops,

cheapest\_price)

def send\_email(msg):

global message

message = MIMEMultipart()

message['Subject'] = 'Current Best flight'

message['From'] = 'myemail@hotmail.com'

message['to'] = 'myotheremail@hotmail.com'

server.sendmail('myemail@hotmail.com', 'myotheremail@hotmail.com', msg)

for i in range(8):

link = 'https://www.expedia.com/'

browser.get(link)

time.sleep(5)

#choose flights only

flights\_only = browser.find\_element\_by\_xpath("//button[@id='tab-flight-tab-hp']")

flights\_only.click()

ticket\_chooser(return\_ticket)

dep\_country\_chooser('Cairo')

arrival\_country\_chooser('New york')

dep\_date\_chooser('04', '01', '2019')

return\_date\_chooser('05', '02', '2019')

search()

compile\_data()

#save values for email

current\_values = df.iloc[0]

cheapest\_dep\_time = current\_values[0]

cheapest\_arrival\_time = current\_values[1]

cheapest\_airline = current\_values[2]

cheapest\_duration = current\_values[3]

cheapest\_stops = current\_values[4]

cheapest\_price = current\_values[-1]

print('run {} completed!'.format(i))

create\_msg()

connect\_mail(username,password)

send\_email(msg)

print('Email sent!')

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

time.sleep(3600)