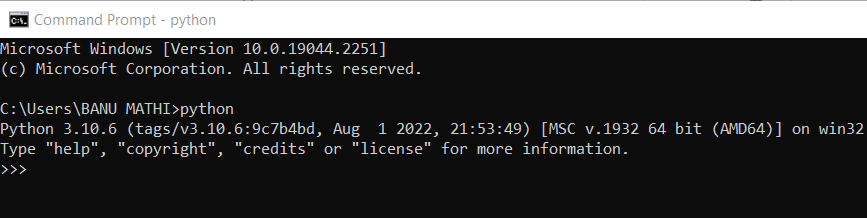
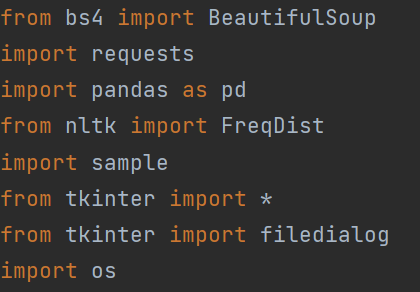
**PACKAGES AND SETUP**

## Procedure:

* 1. Install python. Python version 3.10.6 is used for this project.



* 1. Import the packages



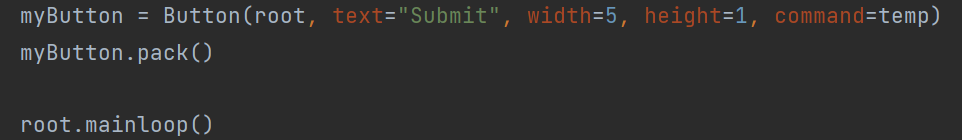
* 1. Write program sample for making proxy request to the target website

Sample.py



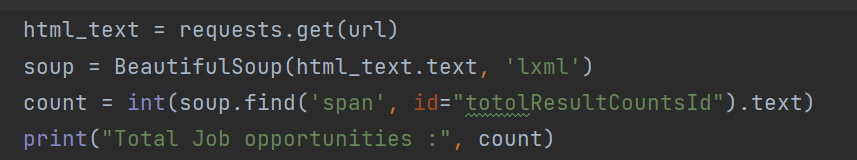
* 1. Get input for familiar skills



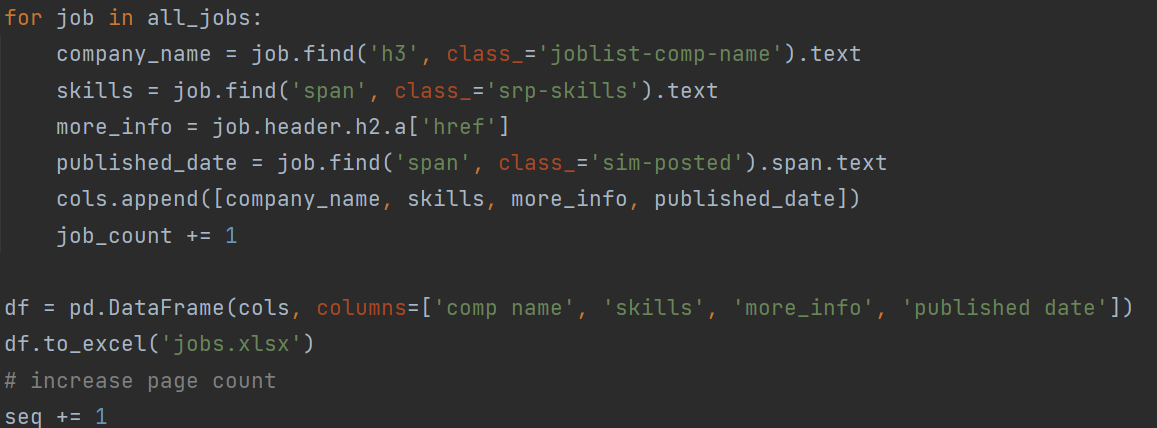


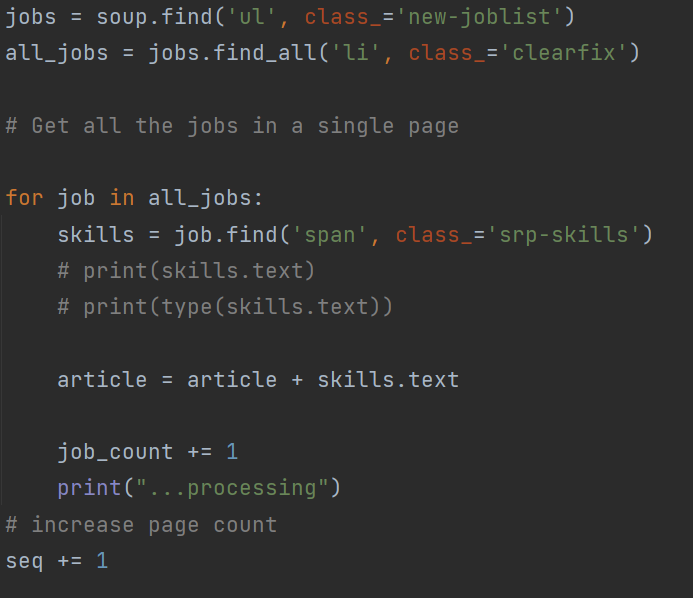
* 1. Generate url and send request

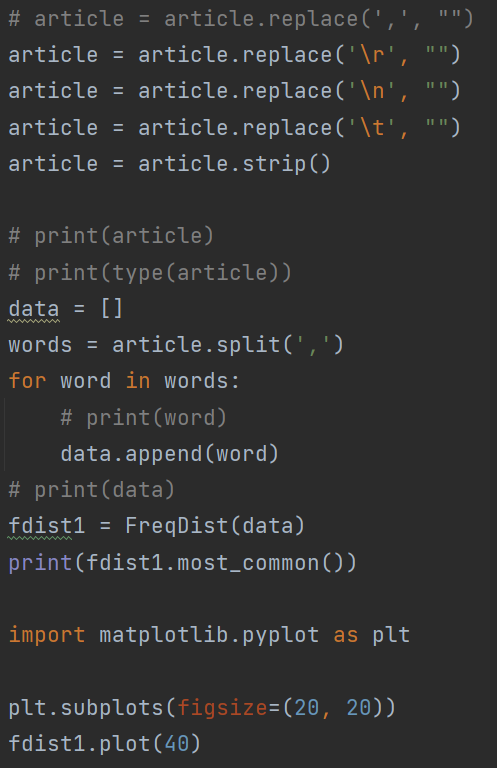


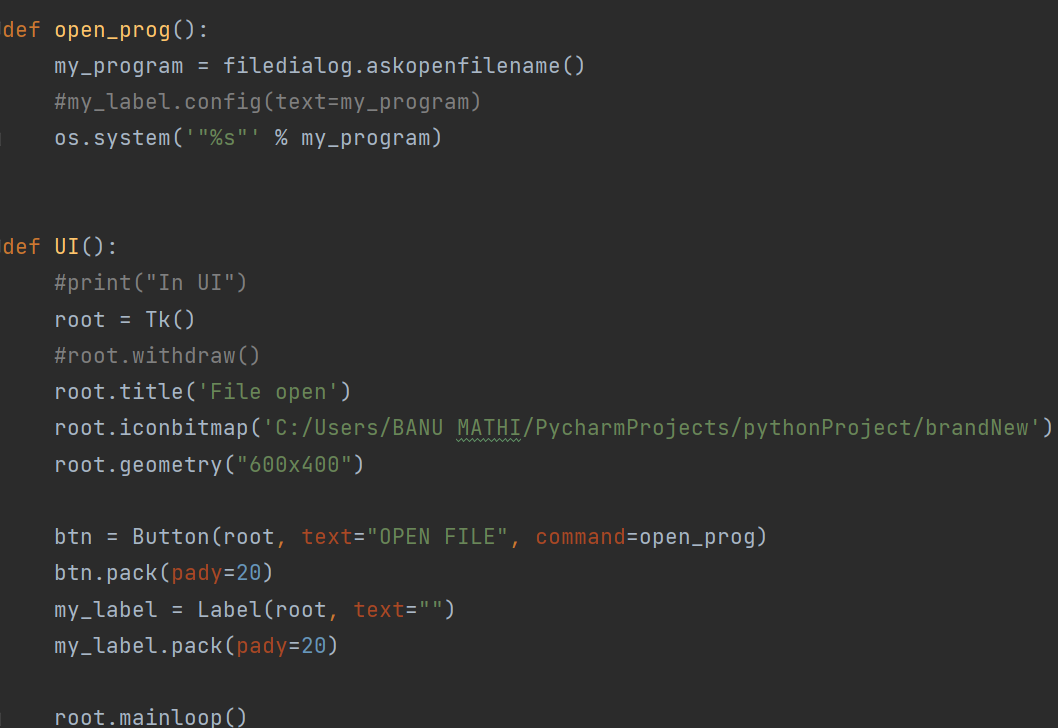
* 1. Create a beautiful soup object and find total number of job opportunities that suit our requirement  
       
     
  2. Get jobs page by page until all the jobs are fetched.

(In this example first 10 pages are scraped)

* 1. Create a function for making a graph on demand of skills based on word count   
     (using python nltk)  
       
       
     



* 1. Create UIs  
       
       
     

Source code:

from bs4 import BeautifulSoup

import requests

import pandas as pd

from nltk import FreqDist

import sample

from tkinter import \*

from tkinter import filedialog

import os

string = ""

data = []

def get\_ip():

root = Tk()

root.geometry("300x300")

root.title('Enter skills')

root.eval('tk::PlaceWindow . center')

myLabel = Label(root, text="Enter your skills", font=("Helvetica", 12))

myLabel.pack()

e = Entry(root, width=50, font=('Helvetica 12'))

e.pack()

def temp():

global data

global string

e.get()

string = f'{e.get()}'

data = string.split()

print(data)

print(type(data))

root.quit()

myButton = Button(root, text="Submit", width=5, height=1, command=temp)

myButton.pack()

root.mainloop()

def open\_prog():

my\_program = filedialog.askopenfilename()

#my\_label.config(text=my\_program)

os.system('"%s"' % my\_program)

def UI():

#print("In UI")

root = Tk()

#root.withdraw()

root.title('File open')

root.iconbitmap('C:/Users/BANU MATHI/PycharmProjects/pythonProject/brandNew')

root.geometry("600x400")

btn = Button(root, text="OPEN FILE", command=open\_prog)

btn.pack(pady=20)

my\_label = Label(root, text="")

my\_label.pack(pady=20)

root.mainloop()

def find\_jobs():

# acquire user skills

# Parse User skill set to search

if(len(data)>1):

for x in range(len(data)):

skillset = "%2C".join([data[x].replace(' ', '+')])

else:

skillset=data[0]

#skillset = "%2C".join([x.replace(' ', '+') for x in data]) if len(data) > 1 else data[0]

seq = 1

# Get total jobs count

url = f'https://www.timesjobs.com/candidate/job-search.html?from=submit&actualTxtKeywords' \

f'={skillset}&searchBy=0&rdoOperator=OR&searchType=personalizedSearch&luceneResultSize=25&postWeek=60&txtKeywords=' \

f'{skillset}&pDate=I&sequence={seq}&startPage=1'

html\_text = requests.get(url)

soup = BeautifulSoup(html\_text.text, 'lxml')

count = int(soup.find('span', id="totolResultCountsId").text)

print("Total Job opportunities :", count)

job\_count = 1

# Get jobs page by page until all the jobs are fetched

# while job\_count <= count:

while seq<=10:

#while seq <= 1:

url = f'https://www.timesjobs.com/candidate/job-search.html?from=submit&actualTxtKeywords=' \

f'{skillset}&searchBy=0&rdoOperator=OR&searchType=personalizedSearch&luceneResultSize=25&postWeek=60&txtKeywords=' \

f'{skillset}&pDate=I&sequence={seq}&startPage=1'

print("Scraping page :", seq)

html\_text = sample.send\_request(url)

soup = BeautifulSoup(html\_text.text, 'lxml')

jobs = soup.find('ul', class\_='new-joblist')

all\_jobs = jobs.find\_all('li', class\_='clearfix')

# Get all the jobs in a single page

cols = []

for job in all\_jobs:

company\_name = job.find('h3', class\_='joblist-comp-name').text

skills = job.find('span', class\_='srp-skills').text

more\_info = job.header.h2.a['href']

published\_date = job.find('span', class\_='sim-posted').span.text

cols.append([company\_name, skills, more\_info, published\_date])

job\_count += 1

df = pd.DataFrame(cols, columns=['comp name', 'skills', 'more\_info', 'published date'])

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

# increase page count

seq += 1

def find\_jobs1():

seq = 1

# Get total jobs count

url = f'https://www.timesjobs.com/candidate/job-search.html?from=submit&searchType=' \

f'Home\_Search&funcAreaSpec=35115&luceneResultSize=25&postWeek=60&cboPresFuncArea=35' \

f'&pDate=Y&sequence={seq}&startPage=11'

html\_text = requests.get(url)

soup = BeautifulSoup(html\_text.text, 'lxml')

count = int(soup.find('span', id="totolResultCountsId").text)

print("Total Job opportunities :", count)

job\_count = 1

# Get jobs page by page until all the jobs are fetched

article = ''

while seq <= 10:

url = f'https://www.timesjobs.com/candidate/job-search.html?searchType=' \

f'Home\_Search&from=submit&asKey=OFF&txtKeywords=&cboPresFuncArea=35'

html\_text = requests.get(url)

soup = BeautifulSoup(html\_text.text, 'lxml')

jobs = soup.find('ul', class\_='new-joblist')

all\_jobs = jobs.find\_all('li', class\_='clearfix')

# Get all the jobs in a single page

for job in all\_jobs:

skills = job.find('span', class\_='srp-skills')

# print(skills.text)

# print(type(skills.text))

article = article + skills.text

job\_count += 1

print("...processing")

# increase page count

seq += 1

# article = article.replace(',', "")

article = article.replace('\r', "")

article = article.replace('\n', "")

article = article.replace('\t', "")

article = article.strip()

# print(article)

# print(type(article))

data = []

words = article.split(',')

for word in words:

# print(word)

data.append(word)

# print(data)

fdist1 = FreqDist(data)

print(fdist1.most\_common())

import matplotlib.pyplot as plt

plt.subplots(figsize=(20, 20))

fdist1.plot(40)

window=Tk()

window.withdraw()

window.title('Job infos')

window.geometry("500x500+20+20")

window.eval('tk::PlaceWindow . center')

lb1 = Label(window, text="Select an option",fg='red', font=("Helvetica", 10))

lb1.config(anchor=CENTER)

lb1.pack()

lb2 = Label(window, text="1-Scrape job info",fg='blue', font=("Helvetica", 10))

#lb2.place(x=80, y=70,anchor=CENTER)

lb2.config(anchor=CENTER)

lb2.pack()

lb3 = Label(window, text="2- View demand of skills",fg='blue', font=("Helvetica", 10))

#lb3.place(x=80, y=90,anchor=CENTER)

lb3.config(anchor=CENTER)

lb3.pack()

lb4 = Label(window, text="0-Exit", anchor=CENTER,fg='blue', font=("Helvetica", 10))

#lb4.place(x=80, y=110,anchor=CENTER)

lb4.config(anchor=CENTER)

lb4.pack()

# btn = Button(window, text="submit", command=get\_ip)

# btn.place(x=200, y=150)

btn1=Button(window, text="1", fg='blue',height=1, width=5,command= lambda:[get\_ip(), find\_jobs(), UI()])

btn1.place(x=180, y=160)

btn2=Button(window, text="2", fg='blue', height=1, width=5,command=find\_jobs1)

btn2.place(x=240, y=160)

btn3=Button(window, text="0", fg='blue',height=1, width=5, command=window.destroy)

btn3.place(x=300, y=160)

window.mainloop()

sample.py

import requests

def send\_request(urlweb):

response = requests.get(

url='https://app.scrapingbee.com/api/v1/',

params={

'api\_key': '0E4PM0ERBHIVC9O4NWLGPE91J3DBWBOVBAJXX4RY4PS5SWGG0WZ3C6ORM1BIAPJ6UDKU1WWTQGLSHB1Z',

'url': urlweb,

'wait': '1000',

'block\_ads': 'True',

'render\_js': 'False'

},

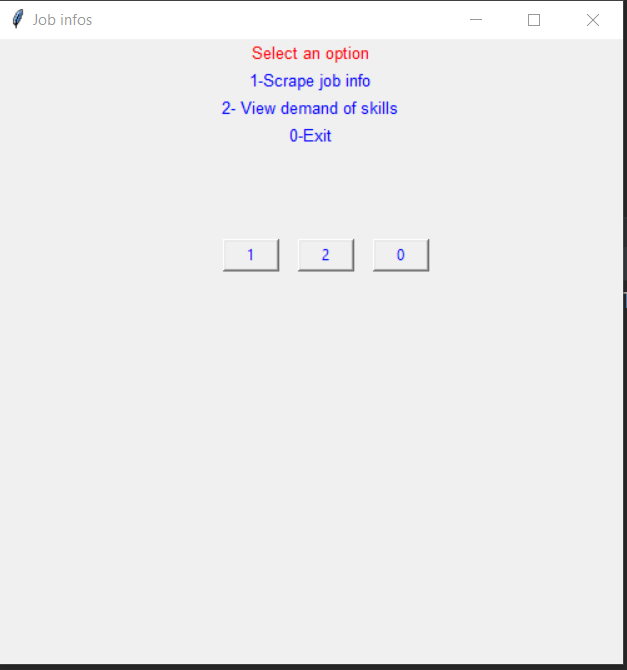
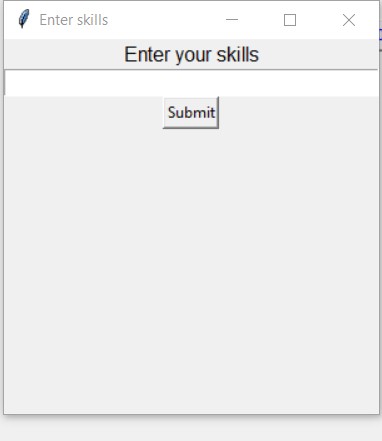
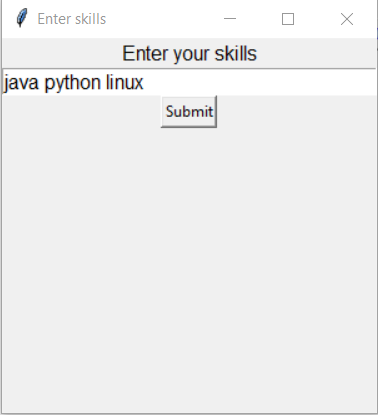
)

# print('Response HTTP Status Code: ', response.status\_code)

# print('Response HTTP Response Body: ', response.content)

return response

**OUTPUT**

**  **

