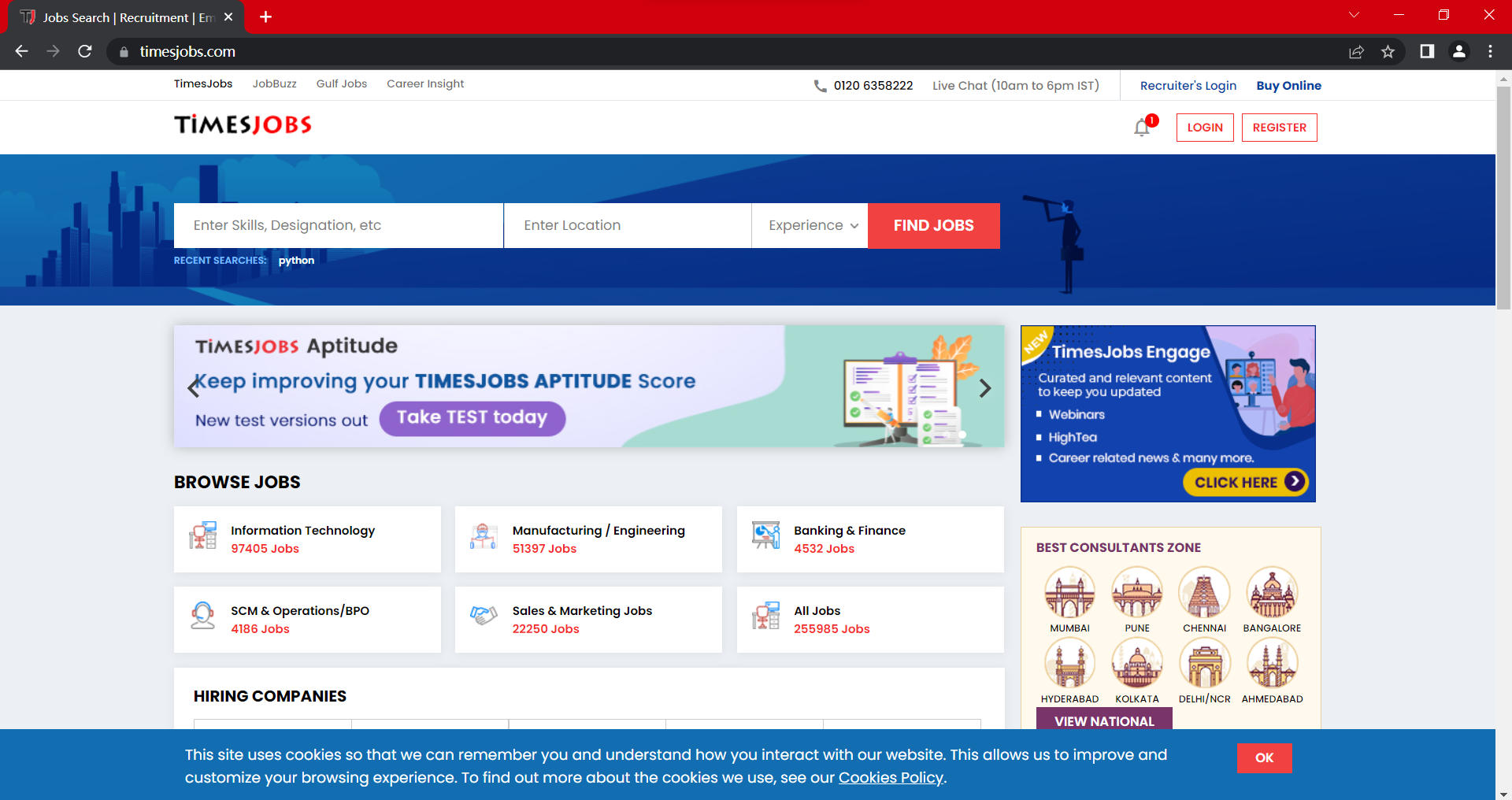
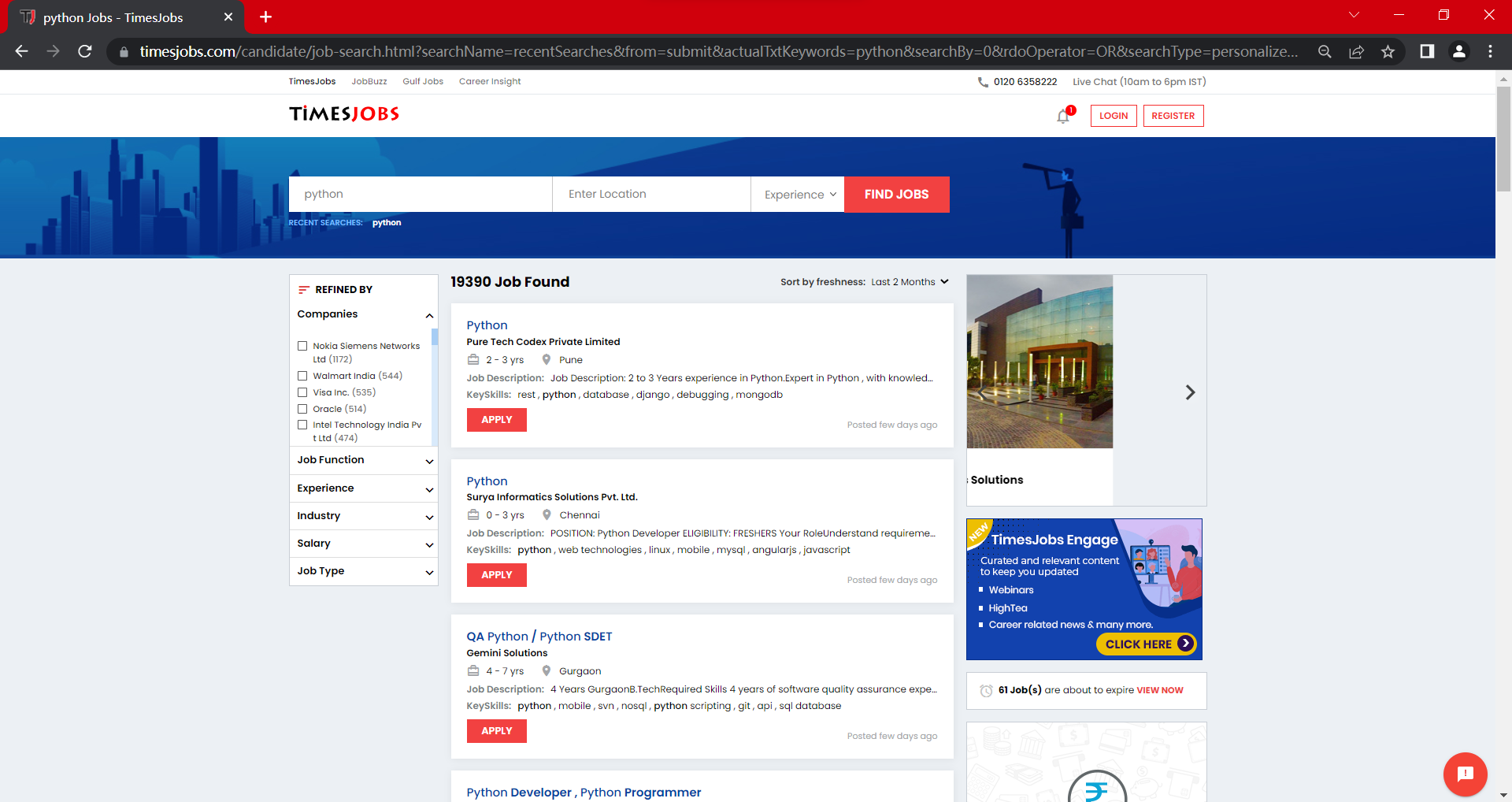
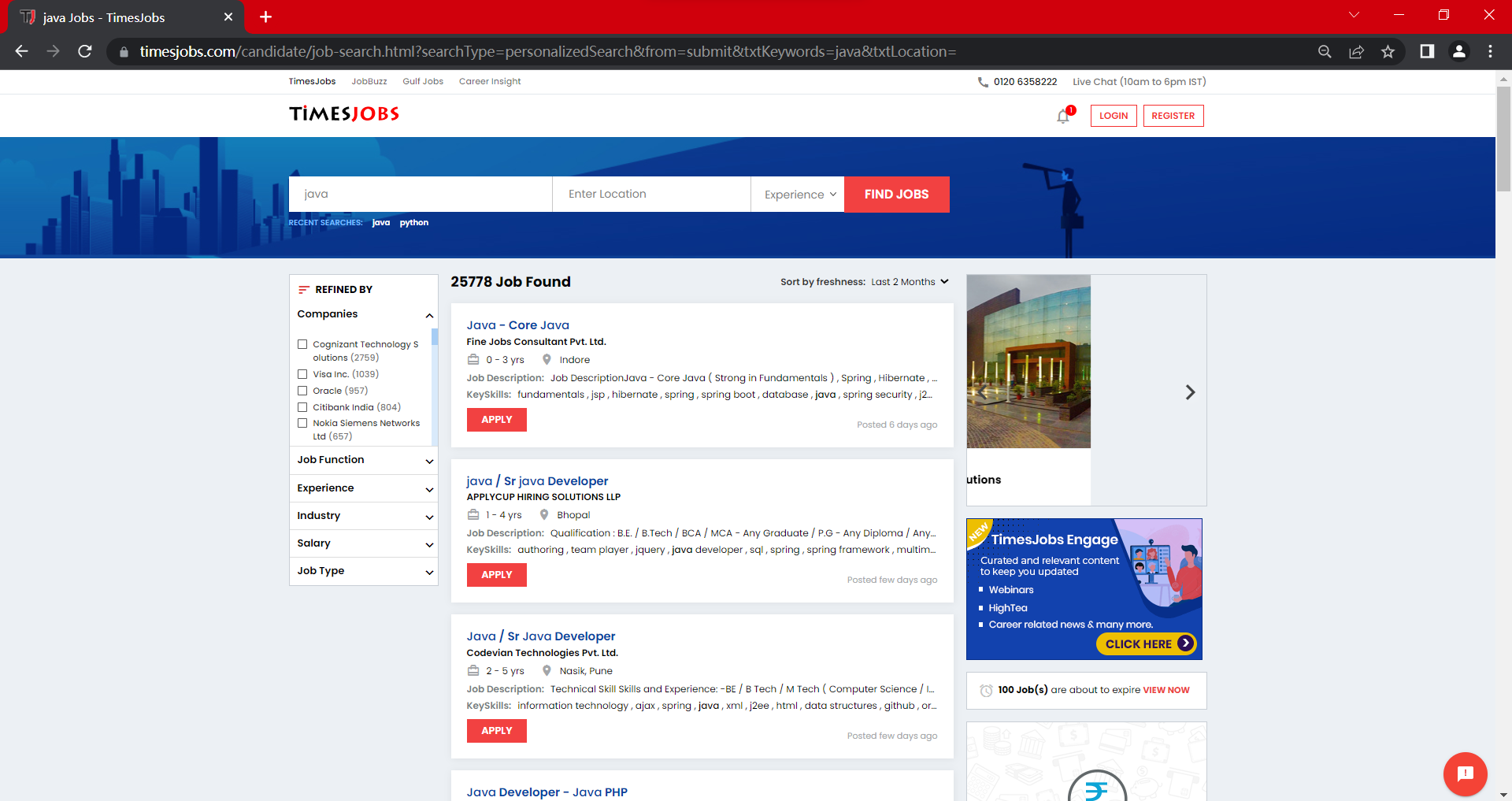
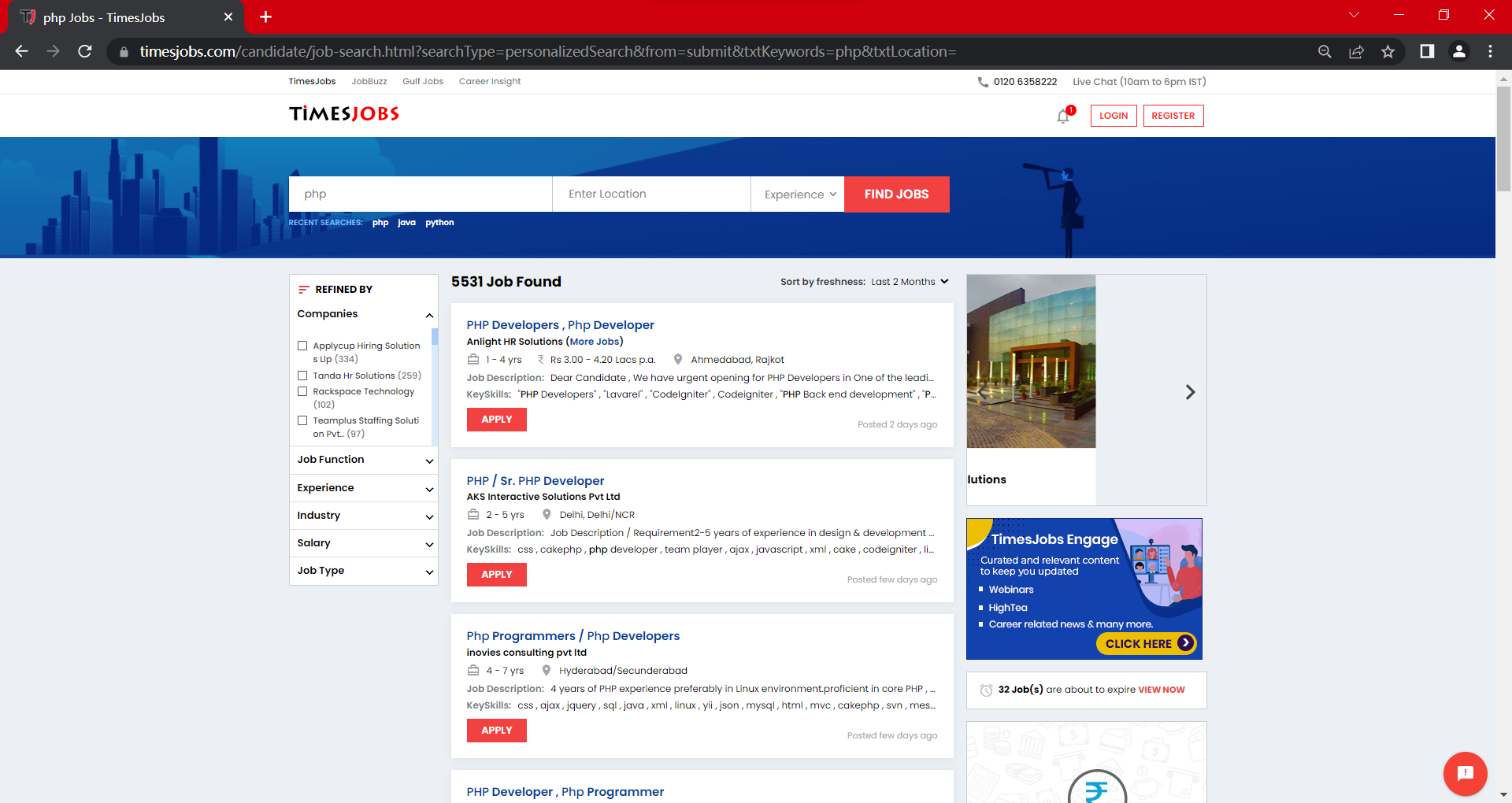
The target website: https://www.timesjobs.com/



The target webpages (search python, java, php):







Python Code:

''' This program will perform a web scrape on TIMESJOBS web pages and extract the following Job information:

job name, company, location, job description, skills, more information (the address of the job web page)'''

*from* bs4 *import* BeautifulSoup

*import* urllib.request

*import* datetime

*# A Job class is created in order to create an object for each job instance and save the information as its attributes.*

class Job:

    def \_\_init\_\_(*self*,*name*, *company*, *location*, *job\_des*, *skills*, *more\_info*):

*self*.name = *name*

*self*.company = *company*

*self*.location = *location*

*self*.job\_des = *job\_des*

*self*.skills = *skills*

*self*.more\_info = *more\_info*

*# This function creates a string include each job's attributes.*

    def get\_string(*self*):

*return* "Job name: " + str(*self*.name) + "\nCompany: " + str(*self*.company) + "\nLocation: " + str(*self*.location) \

            + "\nJob description: " + str(*self*.job\_des) + "\nRequired skills: " + str(*self*.skills) + "\nMore Info: " + *self*.more\_info + "\n\n" + "\*" \* 80 + "\n"

*# This function performs the scrape of TIMESJOB pages. It collets all jobs' information blocks in a page.*

*# It takes one argument: job\_type. The job\_type means the type of job, such as java, php, and python here.*

def req\_data(*job\_type*):

    job\_info = []

*try*:

        source = urllib.request.urlopen(f"https://www.timesjobs.com/candidate/job-search.html?searchType=personalizedSearch&from=submit&txtKeywords={*job\_type*}&txtLocation=")

        tag = BeautifulSoup(source.read(), "html.parser")

        print(f"Requiring {*job\_type*} jobs, please wait a moment...")

*#This statment finds all blocks that includ job information*

        jobs = tag.find\_all('li', *class\_* = 'clearfix job-bx wht-shd-bx')

*#This for loop traverse every single job information block*

*for* job *in* jobs:

            job\_info.append(Job(\*req\_individual\_data(job)))

*return*(job\_info)

*except*:

        print("Your retrieval request was unsuccessful.")

*# This function finds particular information in one block*

*# and assigns each piece of information to a corresponding variable of a job object*

def req\_individual\_data(*job*):

    job\_name = *job*.header.h2.a.text.strip().replace('\r', '')

    company\_name = *job*.find('h3', *class\_* = 'joblist-comp-name').text.replace('(More Jobs)', '').strip().replace('\r', '')

    location = *job*.find('ul', *class\_* = 'top-jd-dtl clearfix').findChild('span').text.strip().replace('\r', '')

    job\_des = *job*.find('ul', *class\_* = 'list-job-dtl clearfix').findChild('li').text.strip().replace('\r', '').replace('\n', '').replace('Job Description:', '')

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

    address = *job*.header.h2.a['href']

*return*(job\_name, company\_name, location, job\_des, skills, address)

*# This function performs a frequency count of all words in the different {job}info.txt file using the dictionary concept.*

def word\_freq(*job\_type*):

*try*:

*with* open(f"{*job\_type*}info.txt", "r") *as* f:

            content = f.read().split()

            job\_dict = {}

*for* word *in* content:

*if* word *in* job\_dict:

                    job\_dict[word] += 1

*else*:

                    job\_dict[word] = 1

            print(f"The frequency of all word occurrences in {*job\_type*}info.txt file is:\n", job\_dict)

*except*:

        print(f"File {*job\_type*}info.txt does not exsit.")

*#This function calls req\_individual\_data function, and writes the retrieved data into a corresponding txt file*

def get\_job\_data(*job\_type*):

*with* open(f"{*job\_type*}info.txt", "w") *as* f:

        f.write('Job Type: ' + *job\_type* + "   Time: " + datetime.datetime.strftime(datetime.datetime.now(), '%Y-%m-%D %H:%M:%S') + \

            "\n\n" + "\*" \* 80 + "\n")

*for* job *in* req\_data(*job\_type*):

*# Calling get\_string method to create strings out of each job's attributes.*

            each\_job = job.get\_string()

            f.write(each\_job)

    print(f"Retrieval {*job\_type*} job information complete. You can find the data in {*job\_type*}info.txt.")

def main():

*#This list stores the types of job that user wants to retrieve.*

    job\_intentions = ['python','java','php']

    print("Retrieving Job information from the TIMESJOBS and saving it in (job\_type)info.txt file...")

*#Retrieve each job type in the job\_intentions list*

*for* job\_type *in* job\_intentions:

        get\_job\_data(job\_type)

*#Checking the frequency of words in {job\_type}info.txt file*

    word\_freq(job\_intentions[0]) *#This statement check pythoninfo.txt*

*#word\_freq(job\_intentions[1]) #This statement check javainfo.txt*

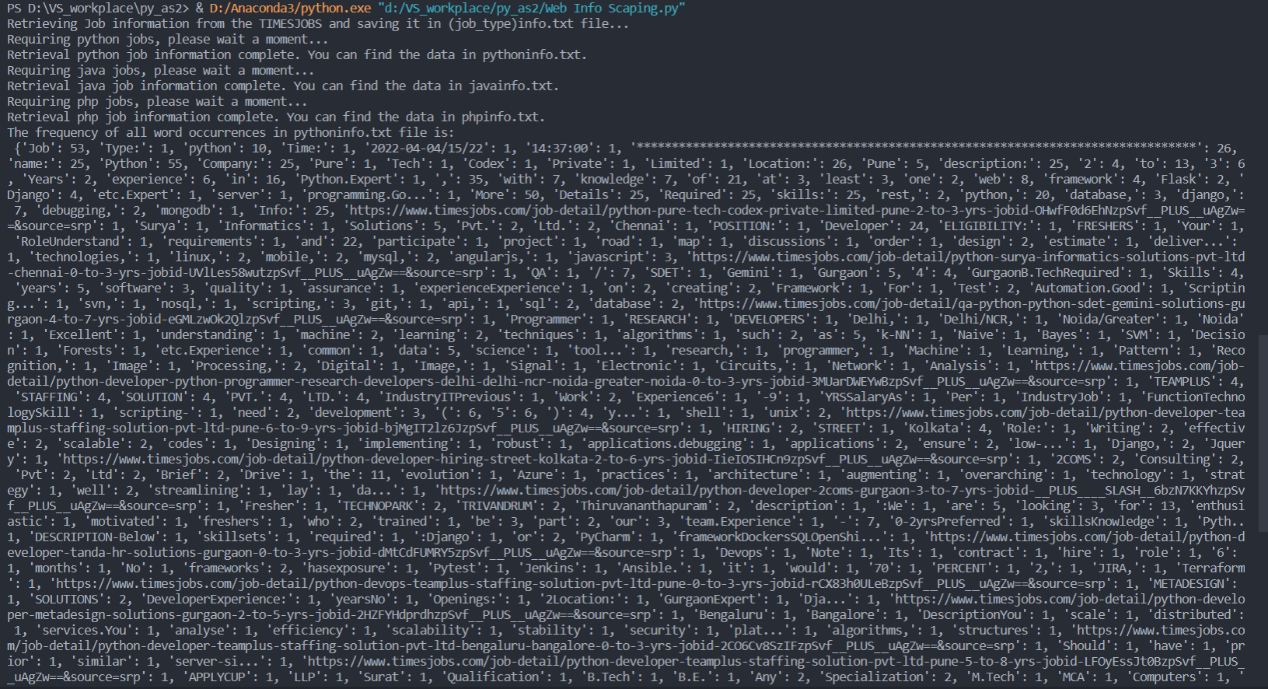
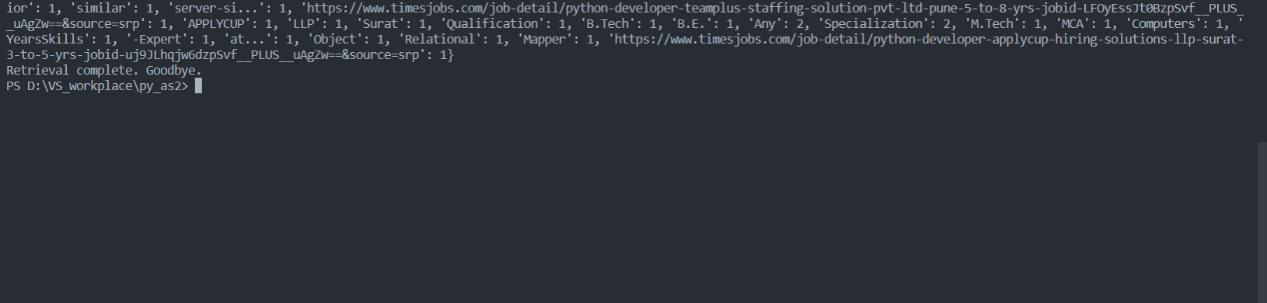
*#word\_freq(job\_intentions[2]) #This statement check phpinfo.txt*

    print("Retrieval complete. Goodbye.")

*if* \_\_name\_\_ == "\_\_main\_\_":

    main()

Terminal Output:



The Result in .txt Files:

