

In [ ]:

```
#Custom
import Functions.GetURLParams as gup
import Functions.Export_to_Excel as ex

#Third Party
import requests
from bs4 import BeautifulSoup
import xlwt

#Built-in
from datetime import date,datetime
from collections import defaultdict
import os

#input Example
'''
1,0,0,0
Web Development
Mumbai,Delhi
2020-12-22
3
1
'''

workbook = xlwt.Workbook()
count = 0

while True:
    count+=1
    final_params = gup.get_URL_params()
    URL = 'https://internshala.com'+final_params.lower()
    page = requests.get(URL)
    soup = BeautifulSoup(page.content, 'html.parser')
    max_pages = int(soup.find(id='total_pages').text.strip())

    limit = int(input("How many pages you would like to get? Max Pages ({max_pages})\n".format(max_pages=max_pages)))
    if limit > max_pages:
        limit = max_pages
        print("Pages Set to Maximum pages present")
    elif limit <= 0:
        limit = 1
```

```

print("Pages set to 1")

flag = 0
if limit > 1:
    flag = input('Different pages on different sheets?(Default: Yes) | 1: No\n')
    if flag == '1':
        sheet = workbook.add_sheet("Sheet - {count}".format(count=count))
        ex.write_header(sheet)
    else:
        flag = '1'
        sheet = workbook.add_sheet("Sheet - {count}".format(count=count))
        ex.write_header(sheet)

params = defaultdict(lambda: [])

for i in range(limit):
    URL += '/page-{i}'.format(i = i+1)
    page = requests.get(URL)
    soup = BeautifulSoup(page.content, 'html.parser')
    if flag != '1':
        sheet = workbook.add_sheet("Sheet - {count}|Page - {i}".format(count=count,i = i+1))
        ex.write_header(sheet)
    intern_titles = soup.find_all(class_ = 'heading_4_5 profile')
    if(len(intern_titles) == 0):
        print('No Results Found...')
        exit()
    print('-----Scraping Page {i} -----'.format(i=i+1))
    for title in intern_titles:
        elem = title.find('a',href=True)
        sub_URL = 'https://internshala.com'+str(elem['href'])

        sub_page = requests.get(sub_URL)
        sub_soup = BeautifulSoup(sub_page.content, 'html.parser')

        params['internship_title'].append(sub_soup.find(class_ = 'profile_on_detail_page').text.strip())
        params['company'].append(sub_soup.find(class_ = 'heading_6 company_name').find('a').text.strip())
        params['location'].append(sub_soup.find(class_ = 'location_link').text.strip())

        info = sub_soup.find(class_ = 'internship_other_details_container')

        other_details = info.find_all(class_ = 'item_body')

```

```

params['duration'].append(other_details[1].text.strip())
params['stipend'].append(other_details[2].text.strip())
params['apply_by'].append(other_details[3].text.strip())
params['applicants'].append(sub_soup.find(class_ = 'applications_message').text.strip())

try :
    skills_raw = sub_soup.find(class_ = 'heading_5_5',string = 'Skill(s) required')
    skills_raw = skills_raw.findNext(class_ = 'round_tabs_container')
    params['skills'].append([str(i.text.strip())+ ' , ' for i in skills_raw.find_all(class_ = 'round_tabs')])
except (IndexError,AttributeError):
    params['skills'].append([])

try :
    perks_raw = sub_soup.find(class_ = 'heading_5_5',string = 'Perks')
    perks_raw = perks_raw.findNext(class_ = 'round_tabs_container')
    params['perks'].append([str(i.text.strip())+ ' , ' for i in perks_raw.find_all(class_ = 'round_tabs')])
except (IndexError,AttributeError):
    params['perks'].append([])

try :
    params['openings'].append(int(sub_soup.find_all(class_='text-container')[-1].text.strip()))
except IndexError:
    params['openings'].append([])
params['link'].append(sub_URL)

if flag != '1':
    ex.write_body(params,sheet)
    params = defaultdict(lambda:[])

if flag == '1':
    ex.write_body(params,sheet)

ex.save_and_export(flag,workbook)

```