In [1]:

```
import requests
from bs4 import BeautifulSoup
from datetime import datetime
import pandas as pd

# FUNCTION TO GENERATE URL
def get_url(position,location):
    position = position.replace(' ','-')
    location = location.replace(' ','-')
    template = 'https://www.jobstreet.com.my/en/job-search/{}-jobs-in-{}/'
    url = template.format(position,location)
    return url

# FUNCTION TO GENERATE ALL CARD INFO
def get_jobs(card):
    job_name = card.find('div','sx2jih0 _2j8fZ_0 sIMFL_0 _1JtWu_0').text
    anchor_tag = card.a
    job_url = 'https://www.jobstreet.com.my'+anchor_tag['href']

    company_name = card.find_all('span','sx2jih0 zcydq82q _18qlyvc0 _18qlyvcv _18qlyvc1
    try:
        if len(company_name)>0:
            company_name = company_name[0]
            comp_name = company_name.text
        else:
            comp_name = 'Company_Confidential'
    except IndexError:
        comp_name = 'Company_Confidential'

    job_location = card.find_all('span','sx2jih0 zcydq82q zcydq810 iwjz4h0')
    try:
        if len(job_location)>1:
            job_location = job_location[1]
            job_loc = job_location.text
        else:
            job_location = job_location[0]
            job_loc = job_location.text
    except IndexError:
        job_loc = 'Not_Available'

    try:
        summary = ''
        tag_text = ''
        job_summary = card.find('ul','sx2jih0 sx2jih3 h6p8rp0 h6p8rp5')
        for eachLI in job_summary("li"):
            tag_text = eachLI.text
            summary = summary + tag_text + ','
        if ',' in summary:
            summary = summary.rstrip(',')
        else:
            summary = 'Not Available'
    except AttributeError:
        summary = 'Not Available'

    job_salary = card.find_all('span','sx2jih0 zcydq82q _18qlyvc0 _18qlyvcv _18qlyvc3 _
    try:
        if len(job_salary)>=2:
            job_salary = job_salary[1]
            salary = job_salary.text
```

```
        else:
            salary = 'Not_Available'
    except IndexError:
        salary = 'Not_Available'

    time_tag = card.time
    post_date = time_tag.get('datetime')
    post_date = post_date.split('T')
    post_date = post_date[0]

    today = datetime.today().strftime('%Y-%m-%d')

    job_info = (job_name,job_url,comp_name,job_loc,summary,salary,
                post_date,today)

    return job_info

# CREATE MAIN FUNCTION FOR INFORMATION REC FROM USER
def main(position,location):
    records = []

    url = get_url(position,location)

    # GENERATING RESPONSE
    response = requests.get(url)

    # CREATING BEAUTIFULSOUP OBJECT
    soup = BeautifulSoup(response.text,'html.parser')

    # FOR THE MAIN FUNCTION TO FIND ALL THE DIV TAG
    cards = soup.find_all('div','sx2jih0 zcydq852 zcydq842 zcydq872 zcydq862 zcydq82a z

    # CALLING FUNCTION FOR ALL CARDS
    for everyCard in cards:
        jobDetails = get_jobs(everyCard)
        records.append(jobDetails)

    # SAVE ALL INFO TO CSV USING PANDAS DATAFRAME

    col = ['Job Name','Job URL','Company Name','Job Location','Summary',
           'Salary','Job Posted on','Current Date']
    job_street_data = pd.DataFrame(records,columns=col)

    job_street_data.to_csv("D:\\csv_savefiles")
```

In [ ]: