Importing important libraries

pandas - To read csv file , to create Data set

numpy - To create numpy array from list

ipywidget - to create upload button

io - to read input csv file

display- to display the csv file , clear_output to clear the screen

In [7]:
```
import pandas as pd
import numpy as np
import ipywidgets as widgets
import io
import ipywidgets as widgets
from IPython.display import display, clear_output
```

Html texts to make output readable

In [8]:
```
text_1 = widgets.HTML(value="<b><h2>Upload your Raw skill and Technical skill file</b> </h2>")
text_2 = widgets.HTML(value="<b><h2>Upload your technical skill file here</b> </h2>")
text_3 = widgets.HTML(value="<b><h2>Cleaned data set is </b> </h2>")
text_3 = widgets.HTML(value="<b><h2>Loading </b> </h2>")
```

function to clean the data and stored that data in global variables Raw_skills , Technical_skills

In this function basically we converted Raw_skills into a numpy array and for each element in numpy array , is it belong in Technical_skills or not , if belongs than we continue else we delete that row

In [9]:
```
def Clean_data():
    global Raw_skills
    global Technical_skills
    raw_head=Raw_skills.columns[0]
    tech_head=Technical_skills.columns[0]
    raw_list=Raw_skills[raw_head].to_numpy()
    i=0
    for skills in raw_list:
        if(Technical_skills[Technical_skills[tech_head]==skills].shape[0]==0):
            Raw_skills=Raw_skills.drop(labels=i,axis=0)
        i=i+1
    display(Raw_skills)
```

This function simpliply convert input Technical skills data set into Data frame and store it into a global variable Technical_skills

try and except part is used to remove the error message before uploading the file

In [10]:
```
def convert_technical_skills(Uploader):
    try:
        input_file=list(Uploader.values())[0]
        content=input_file['content']
        content=io.StringIO(content.decode('utf-8'))
        global Technical_skills
        Technical_skills=pd.read_csv(content)
        display(Technical_skills)
        Clean_data()
    except:
        return
```

here same thing is done for Raw_skills

In [11]:
```
def convert_raw_skills(Uploader):
    try:
        input_file=list(Uploader.values())[0]
        content=input_file['content']
        content=io.StringIO(content.decode('utf-8'))
        global Raw_skills
        Raw_skills=pd.read_csv(content)
        display(Raw_skills)

    except:
        return
```

File upload function used to create button and interact function is used to interact that uploaded file with it's corresponding functions

In [12]:
```
display(text_1)
uploader_1=widgets.FileUpload( accept='.csv', multiple=False )
uploader_2=widgets.FileUpload( accept='.csv', multiple=False )
widgets.interact(convert_raw_skills,Uploader=uploader_1)
widgets.interact(convert_technical_skills,Uploader=uploader_2)
```

Out[12]:
```
<function __main__.convert_technical_skills(Uploader)>
```

In [ ]:

In [ ]:

In [ ]: