```
import time
from bs4 import BeautifulSoup
from seleniumbase import Driver


def main():
    user_keywords = "ERR"
    user_location = "ERR"
    job_title = "ERR"
    job_location = "ERR"
    company_name = "ERR"
    query = "ERR"
    base_url = "ERR"
    apply_link = "ERR"
    search_url = "ERR"
    jobs_html = []
    date_posted = "ERR"
    jobs_list = []
    soup = "ERR"
    data_frame = []
    index  = -1

    # Grab user search parameters
    user_keywords, user_location = get_user_input()

    # Google
    base_url_google = "https://www.google.com/search"
    search_url_google = create_user_search_parameters(user_keywords, user_location,
base_url_google, query)
    soup = get_html_code(search_url_google)
    jobs_list = jobs_list_create_helper(soup, 'tNxQIb PUpOsf')
    find_job_data(soup, jobs_list, 'tNxQIb PUpOsf', 0)
    find_job_data(soup, jobs_list, 'wHYlTd MKCbgd a3jPc', 1)
    find_job_data(soup, jobs_list, 'wHYlTd FqK3wc MKCbgd', 2)
    find_job_data(soup, jobs_list, 'Yf9oye', 3)
    find_job_data(soup, jobs_list, 'nNzjpf-cS4Vcb-PvZLI-Ueh9jd-LgbsSe-Jyewjb-tlSJBe', 4)
    print(jobs_list)
```

```
def get_user_input():
    print("Please enter any keyword that you would like with the spaces being replaced by +")
    print("\tExample: Data+Scientist, Computer+Engineer, etc..")
    user_keywords = input("\tEnter: ")

    print("\nPlease enter any location that you would like with the spaces being replaced by
+")
    print("\tExample: Kearney+Nebraska, New+York, etc...")
    user_location = input("\tEnter: ")

    return user_keywords, user_location


def input_valid(user_keywords, user_location):
    return


def create_user_search_parameters(user_keywords, user_location, base_url_google,
query):
    query = f"?q={user_keywords}+jobs+in+{user_location}&ibp=htl;jobs"
    search_url = base_url_google + query
    print("Search URL:", search_url)
    return search_url


def get_html_code(search_url):
    driver = Driver(browser="Chrome", headless=False)
    driver.get(search_url)
    bottom_height = driver.execute_script("return document.body.scrollHeight")
    while True:
        driver.execute_script("window.scrollTo(0, document.body.scrollHeight);")
        time.sleep(.5)
        new_height = driver.execute_script("return document.body.scrollHeight")
        if new_height == bottom_height:
            soup = BeautifulSoup(driver.page_source, 'html.parser')
            return soup
        bottom_height = new_height
```

```
def jobs_list_create_helper(soup, class_name):
    job_cards = soup.find_all('div', class_=f'{class_name}')
    rows, cols = (len(job_cards), 5)
    jobs_list = [[0 for i in range(cols)] for j in range(rows)]
    return jobs_list

def find_job_data(soup, jobs_list, class_name, index):
    job_cards = soup.find_all('div', class_=f'{class_name}')
    counter = 0
    for equipment_type in job_cards:
        jobs_list[counter][index] = equipment_type.text
        counter += 1
    return jobs_list


def convert_to_csv(jobs_list, data_frame):
    return


if __name__ == '__main__':
    main()
```