Python Automation Assignment

Objective:

The objective is to gather and analyze data from senior living operator and community websites using Python-based web scraping techniques.

1. Importing Required Libraries

In [1]:
```
import requests
from bs4 import BeautifulSoup
import pandas as pd
from selenium import webdriver
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from webdriver_manager.chrome import ChromeDriverManager
import time
```

2. Initializing Selenium WebDriver and Setup Chrome WebDriver

In [2]:
```
def setup_driver():
    options = webdriver.ChromeOptions()
    options.add_argument( '--headless' )
    driver_path = ChromeDriverManager().install()
    driver = webdriver.Chrome( executable_path = driver_path, options = opti
    print( "Driver Setup: Success" )
    return driver
```

3.Scraping Website Data

In [3]:
```
def scrape_website( url, driver ):
    driver.get( url )
    try:
        wait = WebDriverWait( driver, 10 )
        project_list = wait.until( EC.presence_of_element_located( ( By.CLAS
        loc_all_section = driver.find_element_by_id('loc_all')

        project_list = wait.until(EC.presence_of_element_located((By.CLASS_N
        loc_all_section = driver.find_element_by_id( 'loc_all' )
        loc_all_soup = BeautifulSoup(loc_all_section.get_attribute( 'innerHT

        projects = []
        flag = 0

        for item in loc_all_soup.select( 'ul.project-list > li.item' ):
            project_info = item.select_one( 'div.project-tile')

            if project_info:
                name = project_info.select_one( 'h3.project-name' ).text.str
                address = project_info.select_one('address.project-address'
                price_range = project_info.select_one( 'p.project-price' ).
                project_details_url = item.find( 'a' )[ 'href' ]

                projects.append( { 'name': name, 'address': address, 'price_
                                'project_details_url': project_details_url
            else:
                print( f"Project info not found for {item}." )
                flag = 1
                continue

        if flag == 0:
            print( "Fetching Data: Success" )
        return projects

    except Exception as e:
        print( "An error occurred during scraping:", e )
        return []
```

4. Fetching Amenities

In [4]:

```
def fetch_amenities( driver, projects ):
    status = 0
    for project in projects:
        try:
            details_url = project[ 'project_details_url' ]
            driver.get( details_url )

            wait = WebDriverWait( driver, 10 )  # Wait for the page to load

            amenities_section = wait.until( EC.presence_of_element_located(

            project_soup = BeautifulSoup( driver.page_source, 'html.parser'

            # Finding the section containing amenities
            amenities_section = project_soup.find( 'div', id = 'amenities'

            if amenities_section:
                # Extracting the list of amenities
                amenities_list = amenities_section.find_all( 'h3', class_='
                amenities = [ amenity.text.strip() for amenity in amenities_
                project['amenities'] = amenities
            else:
                print( "Amenities section not found for:", project[ 'name'
                project[ 'amenities' ] = None
                status = 1

        except Exception as e:
            print( "An error occurred while fetching amenities for project:'
            print( "Error:", e )
            status = 1

        finally:
            driver.back()

    if status == 0:
        print( "Fetching Amenities: Success" )
    return projects
```

5.Cleaning Data and Exporting to CSV File

```
In [5]: def clean_and_export_data( projects ):
            try:
                df = pd.DataFrame( projects )

                df[ 'amenities' ] = df[ 'amenities' ].apply( lambda x: '; '.join([a
                                                          if isinstance(x, list) else

                df[ 'name' ] = df[ 'name' ].str.strip()
                df[ 'address' ] = df[ 'address' ].str.strip()

                df.fillna('Unknown', inplace=True )

                df[ 'name' ] = df[ 'name' ].str.title()
                df[ 'address' ] = df[ 'address' ].str.title()

                df.sort_values( by='name', inplace=True )

                print( "Data Cleaning: Success" )

                df.to_csv( 'senior_living_projects.csv', index=False )
                print( "Data Exported to 'senior_living_projects.csv'" )

            except Exception as e:
                print( "An error occurred during data cleaning and export:", e )

            finally:
                print( "Quitting Driver..." )
                driver.quit()
```

```
In [6]: if __name__ == "__main__":
            url = 'https://www.ashianahousing.com/senior-living-india'
            driver = setup_driver()
            projects = scrape_website( url, driver )
            projects_with_amenities = fetch_amenities( driver, projects )
            clean_and_export_data( projects_with_amenities )
```

```
Driver Setup: Success
Fetching Data: Success
Fetching Amenities: Success
Data Cleaning: Success
Data Exported to 'senior_living_projects.csv'
Quitting Driver...
```