**Software Documentation**

**Minh Le – BS20DSY033**

**Introduction**

The purpose of this software is to scrape information about undergraduate courses offered in Australia from the website of the Universities Admissions Centre (UAC) and store it in a CSV file. The user enters the name of the course they are interested in and the program retrieves information such as the university offering the course, course code, fees, duration, start time, lowest and highest ATAR, lowest and highest selection rank, assumed knowledge, and prerequisites.

1. Ease of Use: This program makes it easy for students in Australia to find relevant courses from different universities. Students can simply input the course name they are interested in, and the program will provide them with a list of all the available options.
2. Time-saving: This program saves a lot of time that would have been spent going through various university websites in search of the course of interest. With just a few clicks, the program provides all the necessary information, including the course fee, duration, start time, lowest and highest ATARs, campus location, and prerequisites.
3. Accessible: The program is accessible to everyone with an internet connection. Students can access it from their homes or any other location without having to travel long distances to different universities to gather information.
4. Provides In-depth Information: The program provides in-depth information on each course, including its assumed knowledge and prerequisites. This information is essential for students to know if they are eligible for the course.
5. Helps with Decision-making: With all the information provided by the program, students can make informed decisions on the courses they want to pursue. They can compare the different options available and choose the one that best suits their needs.
6. Increases Opportunities: This program opens up more opportunities for students in Australia. By providing a comprehensive list of courses from different universities, it gives students more options to choose from, increasing their chances of finding a course that aligns with their interests and career goals.

Overall, this program makes the process of finding and choosing a course much more efficient and effective. It saves time, provides in-depth information, and helps students make informed decisions that can positively impact their future.

**System Requirements**

* Python 3.x
* Selenium webdriver for Chrome
* Pandas library

**Installation**

* Download and install Python 3.x from <https://www.python.org/downloads/>
* Download and install ChromeWebdriver from <https://chromedriver.chromium.org/>
* Install Selenium webdriver for Chrome by running the following command in the terminal:
  + **pip install selenium**
* Install Pandas library by running the following command in the terminal:
  + **pip install pandas**

**Usage**

This code uses Selenium to scrape data from the UAC (Universities Admissions Centre) website. Specifically, it allows a user to input a course name, filters the search results to show only courses from five specified universities, then scrapes detailed information about each course and saves it in a CSV file.

Here is a more detailed breakdown of the code:

1. The user is prompted to input a course name, which is stored in the **course\_input** variable.
2. The script then imports necessary libraries and opens a Chrome window using the Selenium **webdriver** module.
3. The browser navigates to the UAC course search page and waits for 4 seconds for the page to load.
4. The script searches for a text input field on the page using an XPATH selector, types in the course name, and hits the Enter key.
5. The script then filters the search results to only show courses from five specified universities by clicking on checkboxes next to each university name.
6. If the search results contain more than one page, the script clicks on a "Next" button to load additional pages of results until there are no more pages.
7. For each course listed in the search results, the script clicks on the course link to open a new tab/window and scrape detailed information about the course.
8. The script uses XPATH selectors to extract information such as university name, course name, campus, course code, fees, duration, start time, ATAR requirements, selection rank requirements, and assumed knowledge/prerequisites. If any of this information is not available for a given course, the corresponding field in the resulting dataframe will be filled with **None**.
9. The script closes the tab/window containing the detailed course information and repeats steps 7-8 for each course listed in the search results.
10. Once all courses have been scraped, the script cleans up the data by removing any rows that have a **code** field of **None** (i.e. courses that could not be found on the UAC website) and only keeping courses with a **duration** field that contains "3F" (i.e. courses that are three years full-time).
11. The script saves the resulting dataframe as a CSV file with a filename that reflects the input course name.
12. Finally, the script closes the Chrome window and exits.

Approach:

I approached this task by first identifying the website I needed to scrape for course information, and then analyzing its structure to determine the elements I needed to extract. I used the Selenium web driver to automate the browser and interact with the website, and then extracted the necessary information using XPATH selectors. Finally, I cleaned the data and saved it to a CSV file.

Language:

I used Python programming language to write this program. Python is a widely used language for web scraping due to its simplicity and the availability of many useful libraries for the task. Python is also well suited for data cleaning and analysis, which is necessary after scraping data from websites.

Web Scraping Library:

For web scraping, I used the Selenium web driver, which is a popular choice for automating web browsers. Selenium is particularly useful when interacting with dynamic websites, as it allows for user-like interaction with the site. Additionally, Selenium provides a robust library of XPATH selectors, which I used to extract the necessary course information from the website.

Justification:

I chose Python because it is a widely used language for web scraping, and its many libraries make it well suited for the task. Additionally, Python has a strong community of users and developers who regularly contribute to open-source libraries, which further expands its functionality for web scraping and data analysis.

I chose the Selenium web driver because it allowed me to interact with the dynamic website in a user-like way, which was necessary to extract the necessary information. Additionally, Selenium has a robust XPATH selector library, which I used to extract the necessary course information from the website.

Overall, my approach and choices were based on the requirements of the task and the most efficient and effective means to achieve it.

**Test cases**

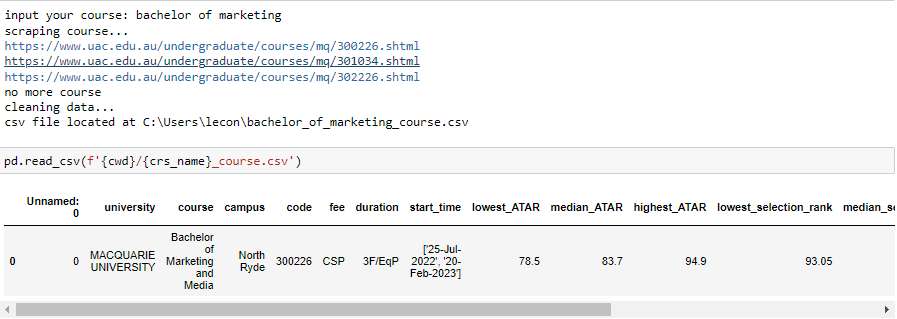
* Test case 1: Input: " bachelor of computer science "

Expected Output: CSV file named "bachelor\_of\_computer\_science\_course.csv" containing data related to undergraduate computer science courses that have a duration of 3 years.



* Test case 2: Input: " bachelor of marketing"

Expected Output: CSV file named " bachelor\_of\_marketing\_course.csv" containing data related to undergraduate marketing courses that have a duration of 3 years.



* Test case 3: Input: " bachelor of civil engineering"

Expected Output: CSV file named " bachelor\_of\_civil\_engineering\_course.csv" containing data related to undergraduate civil engineering courses that have a duration of 3 years.

Graphical user interface, text, application, email

Description automatically generated

**How to execute the file:**

* Download the **zip** file
* Extract the file to your desired directory
* Open the folder
* Locate the **uac\_crawl** application file
* Double click on the file, a command prompt window will appear
* Enter the name of the course you want to scrape information for when prompted.
* The program will open a Chrome window, navigate to the UAC website, and retrieve information about the course.
* The program will then store the information in a CSV file with the name of the course as the filename. The CSV file will be located in the same directory as the program.

**Limitations**

* The program is dependent on the structure of the UAC website and any changes to the website may cause the program to fail.
* The program only retrieves information about undergraduate courses offered in Australia through the UAC website. Other courses or courses offered through other channels will not be included.
* Including ranking and accommodation nearby universities in the code requires scraping data from additional websites that may not be authorized to share data. In addition, these websites have different structures and layouts, making it difficult to extract relevant data automatically. Including this information would require significant additional time and effort to code and test, which may not be feasible given the time limitation of the project.

**Ethical considerations**

* Permission to use data: the program is limited for personal use and is non-profit since the website’ permission to use data was not granted to the student.
* Resource Usage: web scraping can be resource-intensive and can put a significant load on websites. As a result, web scraping can cause a website to slow down or even crash. It is essential to be respectful of the website's resources and avoid any practices that could cause harm.