Web Scraping with Selenium

Objective: The aim of this assignment is to help students understand the basics of web scraping with Selenium, a powerful tool for automating web browser interaction.

Rules: This is an individual assignment. You are allowed to discuss problems and ideas within your group. However, please keep in mind that you are not allowed to share this assignment with other students from your Section or other Sections.

Instructions:

1. Install Selenium. Students should first install Selenium WebDriver for their preferred browser (e.g. Chrome or Firefox). They can do this by following the instructions on the official Selenium website.

2. Choose a website to scrape (see Variants below). Please confirm your variant by providing your name in the following table: click here!

Note: one variant can be assigned to one student only within the same Section.

3. Task 1. Students should use Selenium to write a Java code that will scrape the chosen website. Your program should do the following:

- Open the website in a web browser using Selenium.
- Find and interact with various elements on the page (e.g., links, buttons, text boxes) using Selenium commands.
- Extract data from the page using Selenium commands, such as finding and storing text, images, or other content.
- Save the scraped data in a CSV file or other format of your choice.

4. Task 2. Students need to scrape multiple pages from the same website, or to scrape data from multiple websites and combine the results.

5. Task 3. Students need to use advanced Selenium commands, such as waiting for elements to load or handling pop-up windows.

Variants:

1. Choose a news website and scrape the latest headlines and article summaries.
2. Scrape data from an e-commerce site, such as product names, descriptions, and prices.
3. Scrape data from a job board, such as job titles, locations, and descriptions.
4. Scrape data from a social media site, such as usernames, followers, and posts.
5. Scrape data from a weather site, such as temperature, precipitation, and wind speed.
6. Scrape data from a travel site, such as hotel names, locations, and ratings.
7. Scrape data from a real estate site, such as property addresses, prices, and descriptions.
8. Scrape data from a sports site, such as team names, scores, and schedules.
9. Scrape data from a movie or TV show site, such as titles, ratings, and descriptions.
10. Scrape data from a restaurant site, such as menu items, prices, and reviews.
11. Scrape data from a health and fitness site, such as workout routines, nutrition tips, and health advice.
12. Scrape data from a fashion or beauty site, such as product names, descriptions, and prices.
13. Scrape data from a gaming site, such as game titles, ratings, and descriptions.
14. Scrape data from a music site, such as artist names, albums, and ratings.
15. Scrape data from a stock market site, such as stock prices, volumes, and changes.
16. Scrape data from a government site, such as election results, population statistics, and public records.
17. Scrape data from a cryptocurrency site, such as prices, volumes, and market caps.
18. Scrape data from a news aggregation site, such as headlines and article summaries from multiple sources.
19. Scrape data from a recipe site, such as recipe names, ingredients, and instructions.
20. Scrape data from a book or literature site, such as book titles, authors, and ratings.
21. Scrape data from a fitness tracker site, such as workout logs, steps taken, and calories burned.

22. Scrape data from a podcast site, such as episode titles, descriptions, and ratings.

23. Scrape data from a medical site, such as symptoms, diagnoses, and treatments.

24. Scrape data from a charity or nonprofit site, such as donation amounts, causes, and impact reports.

25. Scrape data from a language learning site, such as vocabulary words, translations, and grammar rules.

26. Scrape data from a financial site, such as loan rates, credit scores, and investment returns.

27. Scrape data from a job review site, such as company ratings, salaries, and interview questions.

28. Scrape data from a gardening or landscaping site, such as plant names, descriptions, and care instructions.

29. Scrape data from a science or technology site, such as research papers, patent filings, and product specifications.

30. Scrape data from a social justice or activism site, such as news articles, petitions, and events.

31. Scrape data from a pet care site, such as breed information, training tips, and product recommendations.

32. Scrape data from a DIY or home improvement site, such as project ideas, tutorials, and supply lists.

33. Scrape data from a university or college site, such as course descriptions, schedules, and faculty bios.

34. Scrape data from a job search site, such as job titles, salaries, and application deadlines.

35. Scrape data from a cybersecurity or privacy site, such as news articles, security threats, and encryption standards.

36. Scrape data from a cultural or historical site, such as museum collections, archives, and exhibits.

37. Scrape data from a law or legal site, such as court rulings, case summaries, and legal news.

38. Scrape data from a language or translation site, such as translation services, dictionaries, and language learning tools.

39. Scrape data from a human resources or talent management site, such as job postings, applicant tracking, and performance reviews.

40. Scrape data from a transportation or logistics site, such as shipping rates, tracking information, and delivery times.

41. Scrape data from a social networking site, such as profiles, friends, and activity feeds.

42. Scrape data from a news aggregator site, such as trending stories, popular articles, and opinion pieces.

43. Scrape data from a legal research site, such as court cases, statutes, and regulations.

44. Scrape data from a medical research site, such as clinical trials, medical journals, and research papers.

45. Scrape data from a fashion and style site, such as clothing items, accessories, and fashion trends.

References:

1. https://www.selenium.dev/documentation/en/

2. https://www.selenium.dev/documentation/en/getting_started_with_webdriver/third_party_drivers_and_plugins/#java

3. https://www.tutorialspoint.com/java_xml/java_xpath_parse_document.htm
https://www.selenium.dev/documentation/en/webdriver/browser_manipulation/#scraping (This link provides an example of how to use Selenium with Java to scrape data from a website. It covers topics such as finding elements on a page, extracting text from those elements, and saving the extracted data to a file).

Submission requirements:

1. You will earn a maximum of 100 points (accounts for 6.7%) for successfully completing this assignment and submitting both your report and source code within the specified deadline.

2. You must submit:

I. A report (in PDF or word), in which you provide the following elements:

- Your task (copy your variant and your task into the report).

- Explanations for the solution provided (explain how you solved the task).

- Outputs (screenshots) with comments and explanations (each screenshot must be numbered (e.g. Fig 1. Displaying misspelled words) and explained what we can see in your screenshot).

II. All Java source code files and classes (in both *.java and *.txt format) needed to run your programs. Your source code must be well-commented. Do not upload your source code to Brightspace as a single zip file. Such submissions will not be accepted.

3. Marks will be deducted if comments/explanations are missing.

4. This assignment is subject to a plagiarism check. The plagiarism check originality score must not exceed 50%. No points will be awarded for assignments submitted via email, Teams, or other platforms, for sending zip archives, or for failing to submit your code in

*.txt files.

5. Assignment submission after the deadline will receive a penalty of 10% for the first 24 hrs, and so on, for up to three days. After three days, the mark will be zero.

6. Unlimited resubmissions are allowed. But keep in mind that we will consider the last

submission. That means that if you resubmit after the deadline, a penalty will be applied, even if you submitted an earlier version on time.

Academic Integrity.

Plagiarism is a serious offense and can result in severe consequences such as receiving a grade of zero on the assignment/lab or even being asked to leave the program.

Copying or using someone else's code is considered plagiarism. This includes using code from online sources, previous labs/assignments, or from other students. Even if you have modified the code, our antiplagiarism software can still show it as plagiarism.

To avoid plagiarism, make sure to always use your own words and ideas when writing source code. Additionally, always check with your instructor to make sure you understand what is allowed and what is not in terms of using outside resources.

Remember that academic integrity is essential for your own personal and professional growth, and it is your responsibility to uphold these principles. So please take it seriously and always produce your own original work.