

Web Scraping Assignment (Medium Level – 1 Hour)

Time Limit: 60 minutes

Difficulty: Medium (but achievable within time)

Experience Level: ~2 Years (RPA / Selenium background)

Objective

Evaluate your ability to scrape structured data using Python, Selenium, and BeautifulSoup, handle pagination, and store clean output in a structured format.

Target Website

<https://quotes.toscrape.com>

Task Description

Scrape quotes data from the website by navigating through the first **3 pages**.

Data to Extract (Mandatory)

- Quote text
- Author name
- Tags (comma-separated)
- Page number from which the quote was scraped

Functional Requirements

1. Use **Python** as the programming language.
2. Use **Selenium** to open the website and navigate between pages.
3. Use **BeautifulSoup** to parse HTML and extract data.
4. Handle pagination for at least 3 pages.
5. Save extracted data into a **CSV file**.
6. Handle basic exceptions (timeouts, missing elements).
7. Use appropriate waits (implicit or explicit).

What You Do NOT Need to Do

- No login handling
- No CAPTCHA solving
- No database storage
- No advanced framework or OOP design

Expected Output

A CSV file with columns:

quote | author | tags | page_number

Suggested Folder Structure (Optional)

```
quote_scraper/  
|-- scraper.py  
|-- output.csv  
|-- README.md
```

README.md (Very Short)

- Python version used
- How to run the script
- Libraries used

Bonus (If Time Permits)

- Parameterize number of pages to scrape
- Explain (in comments) how you would automate this using Automation Anywhere or UiPath

Evaluation Criteria

- Correctness of data extraction
- Proper Selenium + BeautifulSoup usage
- Pagination handling
- Code readability and clarity
- Exception handling approach

Time Guidance

10 min – Inspect HTML

35 min – Write scraping logic

10 min – CSV + cleanup

5 min – README / explanation

Note

Focus on clear logic and working solution rather than perfection.