Finsire

Web Scraping Automation

By Ayush Patel

Introduction

Objective: Automate web scraping process for data extraction.

Tools Used: Selenium WebDriver with Python.

Description of the Website and Data

1. Website: SauceDemo
2. Data Targeted for Scraping: Product inventory information.

Approach

1. Setup Driver:
    - Utilize Selenium WebDriver.
    - Configure headless Chrome browser.
2. Login:
    - Provide username and password.
    - Click on the login button.
3. Scrape Data:
    - Navigate to the inventory page.
    - Extract product names and prices.

Challenges Faced

CAPTCHA Detection: Login failure due to CAPTCHA.

Element Locators: Difficulty in locating dynamic elements.

Scalability: Ensuring the script's scalability for large datasets.

Solutions Implemented

CAPTCHA Handling: Implemented retry mechanism.

Dynamic Element Locators: Utilized explicit waits for element presence.

Scalability: Implemented pagination handling for large datasets.

Insights and Applications

1. Insights:
    - Pricing trends of products.
    - Popular items based on inventory data.
2. Applications:
    - Competitive pricing analysis.
    - Inventory management optimization.

Conclusion

1. Achievements:
    - Successfully automated web scraping process.
    - Obtained valuable inventory data.
2. Future Work:
    - Enhance error handling mechanisms.
    - Implement data analysis pipeline.

Conclusion

Achievements

- Successfully automated web scraping process.
- Obtained valuable inventory data.

Future Work

- Enhance error handling mechanisms.
- Implement data analysis pipeline.