TUNIS BUSINESS SCHOOL
UNIVERSITY OF TUNIS

Cyber Security Project

2nd deliverable

Web Scraping for Assessing Customer Satisfaction on an E-Commerce Site

Participants:

Chammaa Chaima
*ID: ****6736*
Email: chaimachammaa9@gmail.com

Jerbi Siwar
*ID: ****2645*
Email: siwar.jerbi.tbs@gmail.com

Marco Elaa
*ID: ****7089*
Email: ella.marco.tn@gmail.com

Smiri Lina
*ID: ****6262*
Email: lina01sm@gmail.com

IT-360 Information Assurance and Security
Pr. Manel Abdelkader

GitHub Repository: Security Project

Contents

1 Part 1: System Design, Components, and Workflow

Objective: Automatically scrape customer reviews from the product pages, including title, content, rating, reviewer, verification status, date, and brand response across paginated pages, and save the result as a UTF-8 encoded CSV file.

1.1 User Roles

- **Developer:** Runs the script, configures URL, interprets results.
- **Analyst:** Uses the CSV data to analyze customer feedback trends.
- **System:** Selenium WebDriver + Python handles scraping and saving data.

1.2 System Components

1. **Scraper Script (scrape_reviews):**
    - Uses Selenium to load and interact with the site.
    - Parses HTML using BeautifulSoup.
    - Extracts and deduplicates all review data.
    - Handles cookie consent and pagination.
    - Encodes special characters and emojis (UTF-8).
2. **CSV Exporter (save_to_csv):**
    - Cleans product name for filenames.
    - Uses UTF-8-SIG for Excel compatibility.
    - Writes structured data into a CSV with headers.
3. **Web Elements Handled:**
    - Product name
    - Review items
    - Review title
    - Review content
    - Review date
    - Reviewer
    - Rating
    - Brand response
    - Next button

1.3 Working Flow

1. Initialize browser and open product page.
2. Accept cookie popup if shown.
3. Parse current reviews and deduplicate using a unique ID.

4. Extract review fields (title, content, rating, date, etc.).
5. Click the "Next" button to move to the next page.
6. Repeat until all pages are scraped.
7. Save all data into a CSV file.

2 Part 2: Tools & Development Phases

2.1 Tools & Libraries

- Python 3.10+ - Core scripting language.
- Selenium - Browser automation.
- BeautifulSoup - HTML parsing.
- ChromeDriver - For controlling Chrome.
- webdriver-manager - Manages ChromeDriver version.
- csv, re, time - Standard Python libraries.
- PowerBI - For dashboard visualization.

2.2 Development Phases

1. **Phase 1: Prototype Review Scraper (Single Product)**
   - Set up Selenium and scrape one product page.
   - Extract initial reviews using BeautifulSoup.
2. **Phase 2: Pagination Handling**
   - Add logic to click the "Next" button to navigate all pages.
   - Extract reviews across pages.
3. **Phase 3: Data Cleaning & UTF-8 Encoding**
   - Normalize review content.
   - Support special characters and emojis using UTF-8-SIG.
4. **Phase 4: CSV Export**
   - Save structured data with clean filenames.
5. **Phase 5: Automate Multi-Product Scraping**
   - Crawl Decathlon categories to extract all product URLs.
   - Apply scraper to each product.
   - Save reviews in separate CSVs or a database.
6. **Phase 6: Create Dashboard with Insights & KPIs**
   - Using PowerBI to visualize dynamic data.
7. **Phase 8: Automation & Scheduling**
   - Automatically update dashboard with new data.

3 Conclusion

This document outlines the design and components of the proposed web scraping solution for extracting customer reviews from our website product pages. It covers the roles of key users, system components, working flow, and the tools required, including Python, Selenium, and PowerBI. The development phases, from prototype creation to automation, are clearly defined. By following this approach, the solution ensures efficient, scalable scraping and reliable data for analysis, paving the way for successful implementation.

4 References

- Selenium Documentation: https://www.selenium.dev/documentation/
- BeautifulSoup Guide: https://www.crummy.com/software/BeautifulSoup/bs4/doc/
- Decathlon Tunisia: https://www.decathlon.tn