**Project Report: Web Scraping for Product Data Extraction**

**Name: Мұхамедғалым Әділет Олжасұлы**

**ID: 040923550034**

**1. Introduction**

**This report details the execution and results of a Python script designed to perform web scraping. The objective was to programmatically extract structured product information from a target website, process the collected data, and store it in a standard, shareable format (CSV).**

**2. Technologies Utilized**

**The script leverages three essential Python libraries, defining a standard workflow for data extraction and analysis:**

* **requests: Used to send HTTP requests to the target URL and retrieve the raw HTML content of the webpage.**
* **BeautifulSoup (bs4): A powerful library used for parsing the retrieved HTML content. It allows for navigation and searching of the parsed document tree to isolate specific data elements.**
* **pandas: Utilized for creating a robust data structure (DataFrame) to organize the extracted information and for exporting the final dataset to a CSV file.**

**3. Methodology and Execution**

**The script targeted the demonstration e-commerce site, https://books.toscrape.com/, specifically focusing on the first page of the catalogue.**

1. **URL Construction and Request: The script iterates through a defined range of page numbers (in this case, only page 1, since range(1, 2) only includes 1) to construct the target URL: https://books.toscrape.com/catalogue/page-1.html. An HTTP GET request is made using the requests library.**
2. **HTML Parsing: The raw HTML content from the response is passed to BeautifulSoup for parsing.**
3. **Data Isolation: The script first locates the main container for all products (<ol>). It then iterates through every individual product item, identified by the class product\_pod.**
4. **Data Extraction: For each product, the following attributes were extracted:**
   * **Product Name: Extracted from the alt attribute of the <img> tag.**
   * **Rating: Extracted from the second class of the rating <p> tag (e.g., 'Star Rating' class).**
   * **Price: Extracted from the text of the price\_color paragraph, with the currency symbol (£) removed and the result converted to a floating-point number.**
   * **Availability: Extracted from the text content of the instock availability paragraph, ensuring leading/trailing whitespace is removed.**
5. **Data Structuring: The extracted fields (product\_name, product\_price\_range, product\_rating, stock\_status\_availability) are appended as a row to the main list, product\_data.**

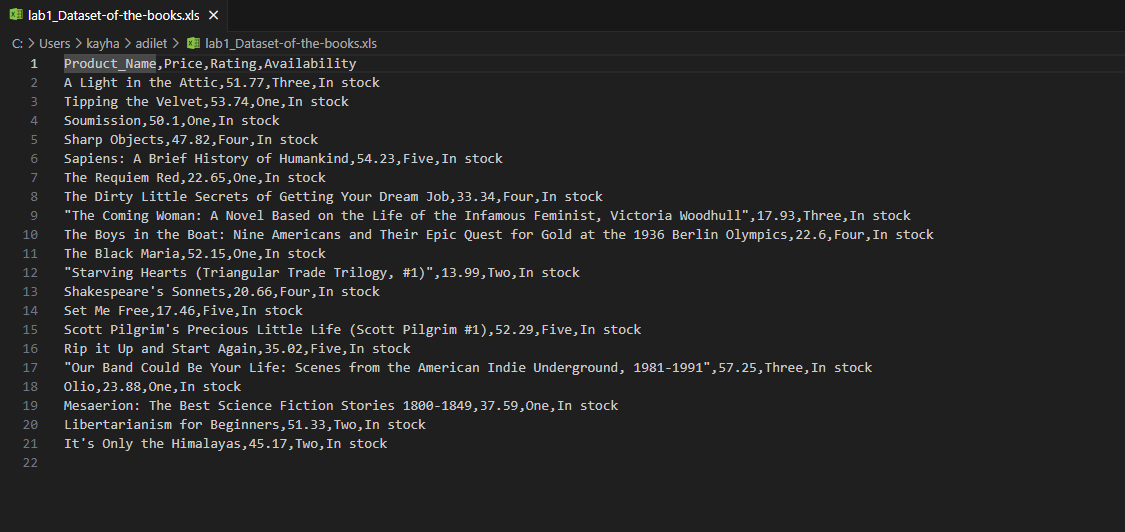
**4. Results and Output**

**Upon completion of the scraping loop, the product\_data list, containing 20 rows of book information, was converted into a Pandas DataFrame (df\_products). The DataFrame was correctly labeled with the column names: 'Product\_Name', 'Price', 'Rating', and 'Availability'.**

**The final dataset was successfully exported to a local file named:**

**lab1\_Dataset-of-the-books.csv**

**This CSV file is ready for subsequent data analysis tasks, providing a structured summary of the product catalogue data extracted.**

****

**import requests**

**from bs4 import BeautifulSoup**

**import pandas as pd**

**product\_data = []**

**for page\_num in range(1, 2):**

**url = f"https://books.toscrape.com/catalogue/page-{page\_num}.html"**

**response = requests.get(url)**

**soup = BeautifulSoup(response.content, 'html.parser')**

**products\_section = soup.find('ol')**

**product\_items = products\_section.find\_all('article', class\_='product\_pod')**

**for item in product\_items:**

**img\_tag = item.find('img')**

**product\_name = img\_tag.attrs['alt']**

**rating\_tag = item.find('p')**

**product\_rating = rating\_tag['class'][1]**

**product\_price\_range = item.find('p', class\_='price\_color').text**

**product\_price\_range = float(product\_price\_range[1:])**

**stock\_status\_availability = item.find('p', class\_='instock availability').text.strip()**

**product\_data.append([product\_name, product\_price\_range, product\_rating, stock\_status\_availability])**

**df\_products = pd.DataFrame(product\_data, columns=['Product\_Name', 'Price', 'Rating', 'Availability'])**

**df\_products.to\_csv('lab1\_Dataset-of-the-books.csv', index=False)**