A picture containing sketch, drawing, embroidery, design

Description automatically generated A picture containing font, text, logo, symbol

Description automatically generated

**Faculty of Computer Science**

Spring Semester 2023

CS 313

**Data Storage and Retrieval**

Submitted to

Dr. Moataz Samy

**Ahmed Nasreldin sehly Norhan salaheldin**

**225359 225257**

**Mariam Adel Yousef Alaa**

**220737 222337**

**Jumia Website**

Extracting Product Information from the Jumia website with Jumia product scrapper.

<https://www.jumia.com.eg/>



**Introduction**

This report discusses using this code, you may scrape product details from a single product page on the Jumia website. An online store called Jumia sells a variety of goods, such as electronics, clothing, cosmetics, and more. A product's name, price, ratings, business name, and URL for the product image can all be gleaned using web scraping techniques.

**Libraries**

The code relies on the following libraries:

**Requests**: Used to transmit HTTP requests and get the web page's HTML content.

A Python module called **BeautifulSoup** makes it simple to extract data from HTML structures by parsing HTML and XML texts.

**csv**: Used to export the data that has been scraped into a CSV file for analysis or storage.

**Importing Required Libraries**: To manage web requests, parse HTML, and put data into a CSV file, we import the required libraries, such as requests, BeautifulSoup, and csv.

**Description of Code**

The code begins by importing the necessary libraries, including requests, BeautifulSoup, and csv for writing the extracted data to a CSV file.

The scrape\_jumia\_product function is defined, which takes a Jumia product URL as an input. Inside the function, the URL is used to make a GET request to the product page using the requests.get method. The response content is then parsed using BeautifulSoup with the 'html.parser' to create a soup object.

The following information is extracted from the product page:

1. **Product Page**: The URL of the product page is stored in the product\_page variable.
2. **Product Name**: The product name element is located using the appropriate HTML class, and the text content is extracted and stored in the product\_name variable.
3. **Price Range**: The price range element is located using the appropriate HTML class, and the text content is extracted and stored in the price\_range variable.
4. **Ratings**: The ratings element is located using the appropriate HTML class, and the text content is extracted and stored in the ratings variable.
5. **Company Name**: The company name element is located using the appropriate HTML class, and the href attribute is extracted and stored in the company\_name variable.
6. **Product Image URL**: The product image element is located using the appropriate HTML class, and the 'data-src' attribute is extracted and stored in the product\_image variable.

The extracted information is then printed to the console for display purposes. Additionally, the data is saved to a CSV file named "jumia\_product.csv" using the csv.DictWriter class. The CSV file includes columns for each extracted data field, and the extracted data is written as a row in the CSV file.

Finally, the product\_url variable is assigned with a sample Jumia product URL, and the scrape\_jumia\_product function is called with this URL as an argument to initiate the scraping process.