

Code Description

web_scrappers

1. fetch_description_amazon.py

- function *description_from_url_amazon(link):*
The function description_from_url_amazon parses c

2. fetch_description_bjs.py

- function *description_from_url_bjs(link):*
The function description_from_url_bjs parses through the link and returns the product details extracted through the link

3. fetch_description_costco.py

- function *description_from_url_costco(link):*
The function description_from_url_costco parses through the link and returns the product details extracted through the link

4. fetch_description_ebay.py

- function *description_from_url_ebay(link):*
The function description_from_url_ebay parses through the link and returns the product details extracted through the link

5. fetch_description_walmart.py

- function *description_from_url_walmart(link):*
The function description_from_url_walmart parses through the link and returns the product details extracted through the link

6. web_scraper.py

- function *get_driver():*
Returns a chrome instance using the Selenium webdriver.
- function *scraper(link):*
Checks which ecommerce platform the link(parameter of function) belongs to and searches for details of similar products on amazon and ebay. Details include description, price and url. This function returns these details as a dictionary.

7. web_scrapper_amazon.py

- function *get_url_amazon(search_term):*
This function returns the Amazon url of the product (search_term) given as an argument.
- function *scrap_amazon(driver, search_term)*
Webpage of the product corresponding to search_term is retrieved using BeautifulSoup and Selenium chrome web driver.
- function *extract_item_amazon(driver, search_term)*

Extracts the product description, price , URL and website name (“Amazon”) best corresponding to search_term and returns it through a dictionary variable.

8. web_scrapper_bjs.py

- function get_url_bjs(search_term)
This function returns the url corresponding to the BJS website for the product (search_term) given as an argument.
- function scrap_bjs(driver, search_term)
Webpage of the product corresponding to search_term is retrieved using BeautifulSoup and Selenium chrome web driver.
- function extract_item_bjs(driver, search_term)
Extracts the product description, price , URL and website name (“BJS”) best corresponding to search_term and returns it through a dictionary variable.

9. web_scrapper_ebay.py

- function get_url_ebay(search_term)
This function returns the url corresponding to the eBay website for the product (search_term) given as an argument.
- function scrap_ebay(driver, search_term)
Webpage of the product corresponding to search_term is retrieved using BeautifulSoup and Selenium chrome web driver.
- function extract_item_ebay(driver, search_term)
Extracts the product description, price , URL and website name (“EBay”) best corresponding to search_term and returns it through a dictionary variable.

10. web_scrapper_walmart.py

- function get_url_walmart(search_term)
This function returns the url corresponding to the walmart website for the product (search_term) given as an argument.
- function scrap_walmart(driver, search_term)
Webpage of the product corresponding to search_term is retrieved using BeautifulSoup and Selenium chrome web driver.
- function extract_item_walmart(driver, search_term)
Extracts the product description, price , URL and website name (“Walmart”) best corresponding to search_term and returns it through a dictionary variable.

extension

11. **index.html**

Default page of an extension that opens when the extension is clicked.

12. **main.css**

This cascading style sheet (CSS) file provides format and style for the contents of the extension display.

13. **manifest.json**

Config file of an extension that provides important information regarding the extension.

14. **popup.js**

This popup.js file fetches the url of the current tab and passes to the server. The response is generated from the server which fetches the same product from different websites and displays it on the extension.

15. **robot.txt**

A robots.txt file tells search engine crawlers which URLs the crawler can access on your site.