

# **Code Description**

## **web\_scrappers**

### **1. FetchDescription.py**

- function fetch\_desc\_amazon  
The function fetch\_desc\_amazon parses the link and gets the product description
- function fetch\_desc\_walmart  
The function fetch\_desc\_walmart parses the link and gets the product description
- function fetch\_desc\_ebay  
The function fetch\_desc\_ebay parses the link and gets the product description
- function fetch\_desc\_costco  
The function fetch\_desc\_costco parses the link and gets the product description
- function fetch\_desc\_bestbuy  
The function fetch\_desc\_bestbuy parses the link and gets the product description
- function fetch\_desc\_bjs  
The function fetch\_desc\_bjs parses the link and gets the product description

### **6. WebScraper.py**

- function scrapper(product\_description):  
Checks for each ecommerce platform the product\_description(parameter of function) for details. 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.
- function run(search\_term)  
Extracts the product description, price , URL and website name 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 bjs url of 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.
- function run(search\_term)  
Extracts the product description, price , URL and website name corresponding to search\_term and returns it through a dictionary variable.

### **9. web\_scrapper\_ebay.py**

- function get\_url\_amazon(search\_term):  
This function returns the Amazon url of the product (search\_term) given as an argument.
- function scrap\_ebay( search\_term)  
Webpage of the product corresponding to search\_term is retrieved using BeautifulSoup.
- function run(search\_term)  
Extracts the product description, price , URL and website name 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 Walmart url of the product (search\_term) given as an argument.
- function scrap\_walmart search\_term)  
Webpage of the product corresponding to search\_term is retrieved using BeautifulSoup.
- function run(search\_term)  
Extracts the product description, price , URL and website name corresponding to search\_term and returns it through a dictionary variable.

### **11. web\_scrapper\_costco.py**

- function get\_url\_costco(search\_term):  
This function returns the Costco url of the product (search\_term) given as an argument.
- function scrap\_costco( search\_term)  
Webpage of the product corresponding to search\_term is retrieved using BeautifulSoup.
- function run(search\_term)

Extracts the product description, price , URL and website name corresponding to search\_term and returns it through a dictionary variable.

## **12.web\_scrapper\_Bestbuy.py**

- function get\_url\_bestbuy(search\_term):  
This function returns the Amazon url of the product (search\_term) given as an argument.
- function scrap\_bestbuy( search\_term)  
Webpage of the product corresponding to search\_term is retrieved using BeautifulSoup.
- function run(search\_term)  
Extracts the product description, price , URL and website name corresponding to search\_term and returns it through a dictionary variable.