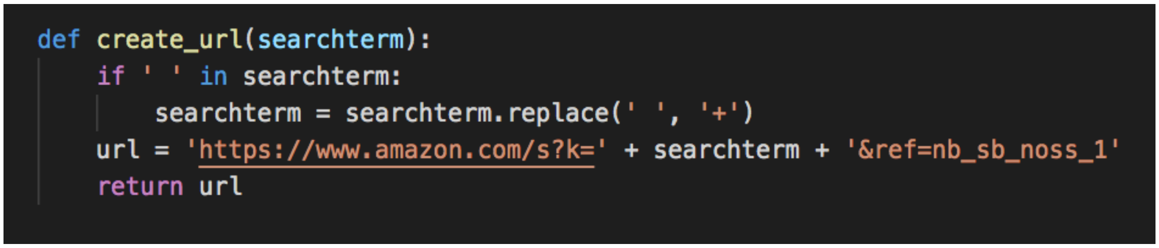
Final Project Data Checkpoint
Munjot Singh

Project Code:

- https://github.com/munjotks/FinalProject-Munjotks.git

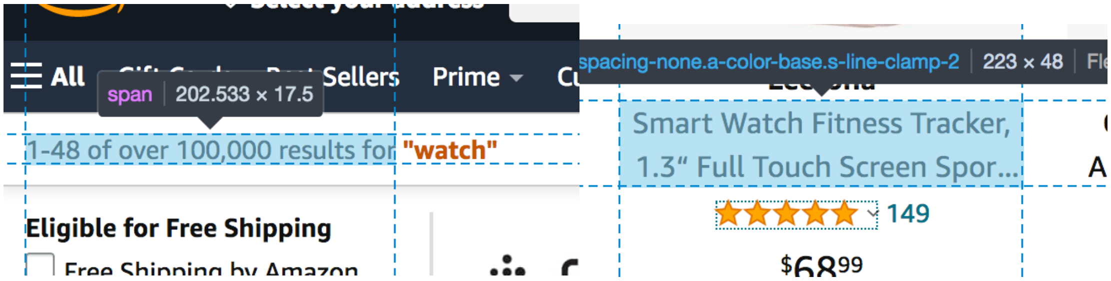
Data Sources:

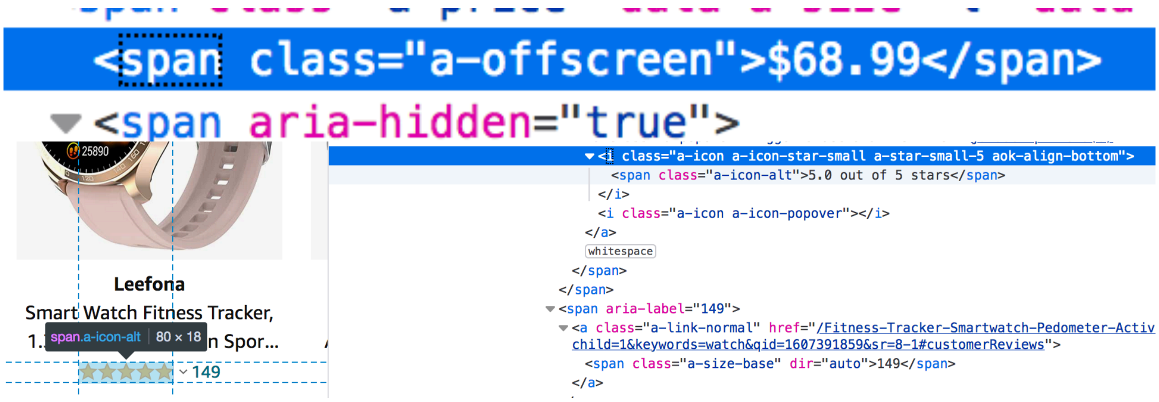
- User will be searching for an Amazon product through the user interface. In order to retrieve the information from amazon, I will be scraping Amazon using cache → https://www.amazon.com/
- The URL for what page to scrape will be generated from https://www.amazon.com/s?k=(SEARCHTERM)&ref=nb_sb_noss_1 and that specific page will be scraped.



```
def create_url(searchterm):
    if ' ' in searchterm:
        searchterm = searchterm.replace(' ', '+')
    url = 'https://www.amazon.com/s?k=' + searchterm + '&ref=nb_sb_noss_1'
    return url
```

- The data I will be collecting from the specific pages will be
    - Search Term | Product Name | # of star out of 5 | Product Price | # of Reviews | # of results

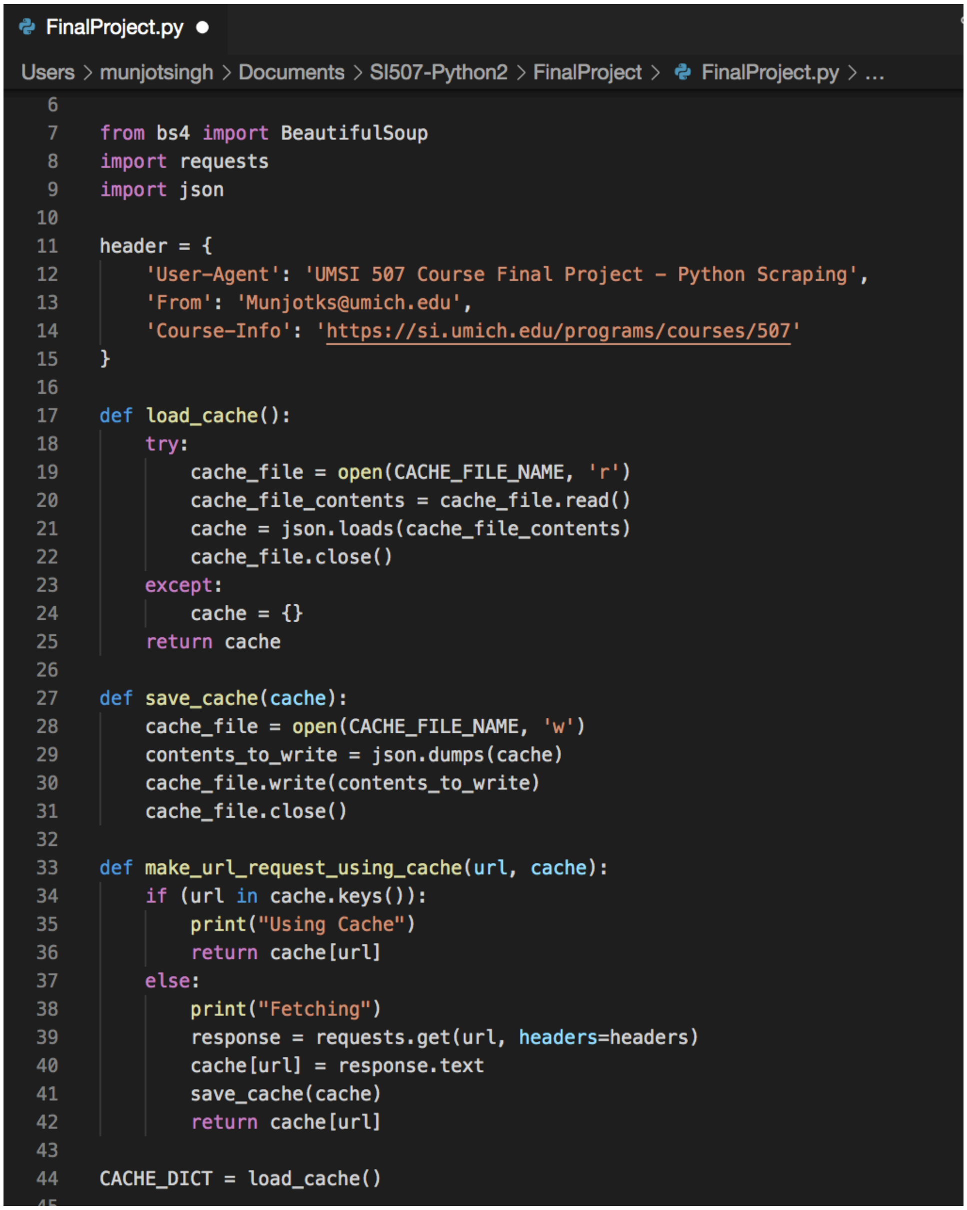

All
Sellers
Prime
span
202.533 × 17.5
1-48 of over 100,000 results for "watch"
Eligible for Free Shipping
Free Shipping by Amazon
spacing-none.a-color-base.s-line-clamp-2
223 × 48
Smart Watch Fitness Tracker,
1.3" Full Touch Screen Spor...
149
$68.99



```
<span class="a-offscreen">$68.99</span>
<span aria-hidden="true">
```

```
<i class="a-icon a-icon-star-small a-star-small-5 aok-align-bottom">
  <span class="a-icon-alt">5.0 out of 5 stars</span>
</i>
<i class="a-icon a-icon-popover"></i>
</a>
whitespace
</span>
</span>
<span aria-label="149">
<a class="a-link-normal" href="/Fitness-Tracker-Smartwatch-Pedometer-Activ
child=1&keywords=watch&qid=1607391859&sr=8-1#customerReviews">
  <span class="a-size-base" dir="auto">149</span>
</a>
```

- Caching will be used every time the same search term is used in the user interface (Same URL)



FinalProject.py ●
Users > munjotsingh > Documents > SI507-Python2 > FinalProject > FinalProject.py > ...

```python

from bs4 import BeautifulSoup
import requests
import json

header = {
    'User-Agent': 'UMSI 507 Course Final Project - Python Scraping',
    'From': 'Munjotks@umich.edu',
    'Course-Info': 'https://si.umich.edu/programs/courses/507'
}

def load_cache():
    try:
        cache_file = open(CACHE_FILE_NAME, 'r')
        cache_file_contents = cache_file.read()
        cache = json.loads(cache_file_contents)
        cache_file.close()
    except:
        cache = {}
    return cache

def save_cache(cache):
    cache_file = open(CACHE_FILE_NAME, 'w')
    contents_to_write = json.dumps(cache)
    cache_file.write(contents_to_write)
    cache_file.close()

def make_url_request_using_cache(url, cache):
    if (url in cache.keys()):
        print("Using Cache")
        return cache[url]
    else:
        print("Fetching")
        response = requests.get(url, headers=headers)
        cache[url] = response.text
        save_cache(cache)
        return cache[url]

CACHE_DICT = load_cache()
```

Database:

- I will be creating a database from the information collected from scraping the search term pages. My two tables will consist of the following fields.
    - Product Table
        - Search Term Category | Product Name | # of Stars | # of Reviews
    - Category (search term) Table
        - Search Term Category | # of Results

EXAMPLE:

Search Term Category	Product Name	Product Price	# of Stars	# of Reviews
Camera	Fujifilm Instax Mini 11 Instant Camera - Lilac Purple	69.00	4.8	2,528
Camera	Digital Camera, Lecran FHD 1080P 36.0 Mega Pixels Vlogging Camera with 16X Digital Zoom, LCD Screen, Compact Portable	48.99	4.3	83
Camera	All-new Blink Outdoor – wireless, weather-resistant HD security camera with two-year battery life and motion detection	59.99	4.3	4,619
Camera	Digital Camera, Lecran FHD 1080P 36.0 Mega Pixels Vlogging Camera	82.98	4.2	83

EXAMPLE:

Search Term Category	# of Results
Camera	10,000+
Pen	3,000

Interaction and Presentation Plans

User will be asked:

- What would you like to search on Amazon?
- User Response: Camera
- How would you like the results displayed?

- User Response: top 10 by stars
- Displays top 10 results
- User can request a scatterplot;
    - Product price vs. # of stars
    - Product price vs. # of reviews