Software Engineering 501
aaakrit@ncsu.edu

Documentation on how the new version is better than the older version:

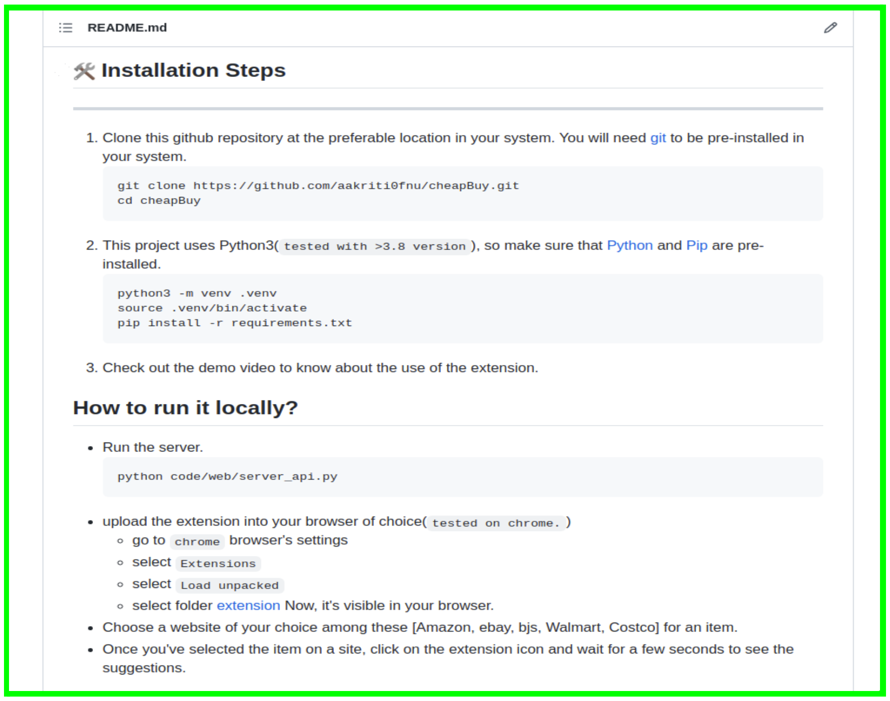
There were mainly 3 areas where the new version is an improvement over the older version

1. **Code Refactoring to make it more comprehensible**
2. **Addition of two new web scrapers(Feature enhancement)- BJ's and Costco**
3. **Docstring and description for each function in the code and Test cases.**

Code Refactoring:

1. **Readme -**
   - Instruction to run the code locally is added – the readme from phase 1 did not consist of running instructions, it only had installation instructions but did not provide any info on how to run the code and which module to run. Added the proper run instructions
   - Extension loading instructions in the browser are added – As the project is a web extension but the instructions to use the extension and unpacking it was missing in the readme. Added the instruction on how to use the extension and how to unload it for running the project
   - The dependency installation written in Readme doesn't work as it is. I need to do certain modifications to make the written installation step work for the project.

Updated Readme.md


README.md

🛠 Installation Steps

1. Clone this github repository at the preferable location in your system. You will need git to be pre-installed in your system.

```
git clone https://github.com/aakriti0fnu/cheapBuy.git
cd cheapBuy
```

2. This project uses Python3(`tested with >3.8 version`), so make sure that Python and Pip are pre-installed.

```
python3 -m venv .venv
source .venv/bin/activate
pip install -r requirements.txt
```

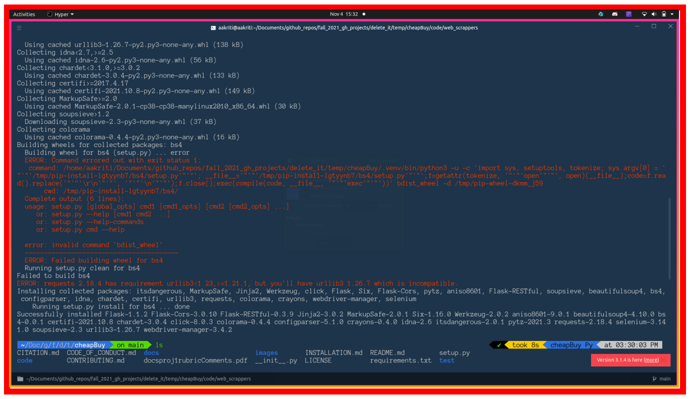
3. Check out the demo video to know about the use of the extension.

How to run it locally?

- Run the server.

```
python code/web/server_api.py
```

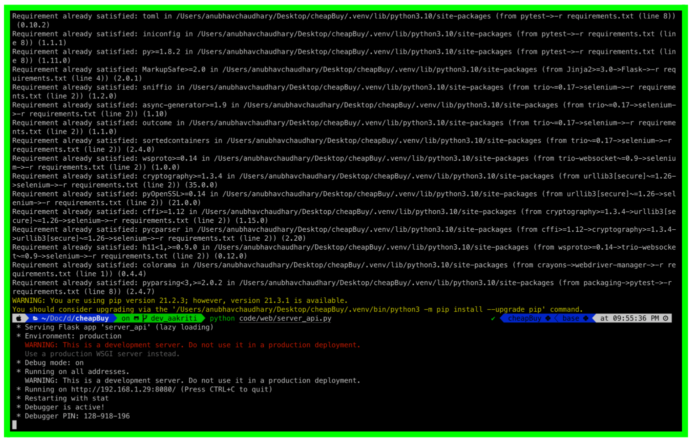
- upload the extension into your browser of choice(`tested on chrome.`)
  - go to `chrome` browser's settings
  - select `Extensions`
  - select `Load unpacked`
  - select folder extension Now, it's visible in your browser.
- Choose a website of your choice among these [Amazon, ebay, bjs, Walmart, Costco] for an item.
- Once you've selected the item on a site, click on the extension icon and wait for a few seconds to see the suggestions.



Activities
Hyper
Nov 4 15:32
aakriti@aakriti:~/Documents/github_repos/fall_2021_gh_projects/delete_it/temp/cheapBuy/code/web_scrappers

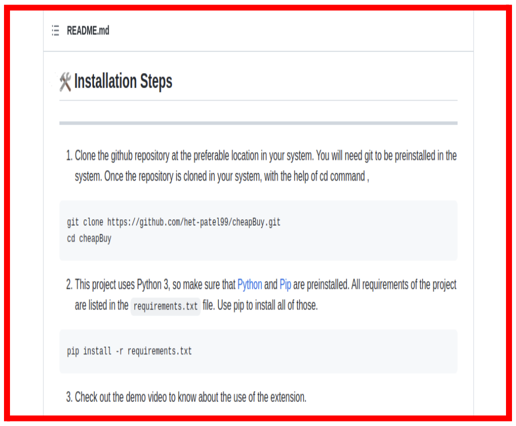
```
  Using cached urllib3-1.26.7-py2.py3-none-any.whl (138 kB)
Collecting idna<2.7,>=2.5
  Using cached idna-2.6-py2.py3-none-any.whl (56 kB)
Collecting chardet<3.1.0,>=3.0.2
  Using cached chardet-3.0.4-py2.py3-none-any.whl (133 kB)
Collecting certifi>=2017.4.17
  Using cached certifi-2021.10.8-py2.py3-none-any.whl (149 kB)
Collecting MarkupSafe>=2.0
  Using cached MarkupSafe-2.0.1-cp38-cp38-manylinux2010_x86_64.whl (30 kB)
Collecting soupsieve>1.2
  Downloading soupsieve-2.3-py3-none-any.whl (37 kB)
Collecting colorama
  Using cached colorama-0.4.4-py2.py3-none-any.whl (16 kB)
Building wheels for collected packages: bs4
  Building wheel for bs4 (setup.py) ... error
  ERROR: Command errored out with exit status 1:
   command: /home/aakriti/Documents/github_repos/fall_2021_gh_projects/delete_it/temp/cheapBuy/.venv/bin/python3 -u -c 'import sys, setuptools, tokenize; sys.argv[0] = '
"'"'/tmp/pip-install-lgtyynb7/bs4/setup.py'"'"'; __file__='"'"'/tmp/pip-install-lgtyynb7/bs4/setup.py'"'"';f=getattr(tokenize, '"'"'open'"'"', open)(__file__);code=f.rea
d().replace('"'"'\r\n'"'"', '"'"'\n'"'"');f.close();exec(compile(code, __file__, '"'"'exec'"'"'))' bdist_wheel -d /tmp/pip-wheel-dkmm_j59
       cwd: /tmp/pip-install-lgtyynb7/bs4/
  Complete output (6 lines):
  usage: setup.py [global_opts] cmd1 [cmd1_opts] [cmd2 [cmd2_opts] ...]
     or: setup.py --help [cmd1 cmd2 ...]
     or: setup.py --help-commands
     or: setup.py cmd --help

  error: invalid command 'bdist_wheel'
  ----------------------------------------
  ERROR: Failed building wheel for bs4
  Running setup.py clean for bs4
Failed to build bs4
ERROR: requests 2.18.4 has requirement urllib3<1.23,>=1.21.1, but you'll have urllib3 1.26.7 which is incompatible.
Installing collected packages: itsdangerous, MarkupSafe, Jinja2, Werkzeug, click, Flask, Six, Flask-Cors, pytz, aniso8601, Flask-RESTful, soupsieve, beautifulsoup4, bs4,
 configparser, idna, chardet, certifi, urllib3, requests, colorama, crayons, webdriver-manager, selenium
    Running setup.py install for bs4 ... done
Successfully installed Flask-1.1.2 Flask-Cors-3.0.10 Flask-RESTful-0.3.9 Jinja2-3.0.2 MarkupSafe-2.0.1 Six-1.16.0 Werkzeug-2.0.2 aniso8601-9.0.1 beautifulsoup4-4.10.0 bs
4-0.0.1 certifi-2021.10.8 chardet-3.0.4 click-8.0.3 colorama-0.4.4 configparser-5.1.0 crayons-0.4.0 idna-2.6 itsdangerous-2.0.1 pytz-2021.3 requests-2.18.4 selenium-3.14
1.0 soupsieve-2.3 urllib3-1.26.7 webdriver-manager-3.4.2

 ~/Doc/g/f/d/t/cheapBuy  on main  ls                                                   ✓  took 8s  cheapBuy Py  at 03:30:03 PM
CITATION.md   CODE_OF_CONDUCT.md  docs                      images       INSTALLATION.md  README.md         setup.py
code          CONTRIBUTING.md     docsproj1rubricComments.pdf  __init__.py  LICENSE          requirements.txt  test
```

Version 3.1.4 is here (more)
~/Documents/github_repos/fall_2021_gh_projects/delete_it/temp/cheapBuy/code/web_scrappers
main



```
Requirement already satisfied: toml in /Users/anubhavchaudhary/Desktop/cheapBuy/.venv/lib/python3.10/site-packages (from pytest->-r requirements.txt (line 8))
 (0.10.2)
Requirement already satisfied: iniconfig in /Users/anubhavchaudhary/Desktop/cheapBuy/.venv/lib/python3.10/site-packages (from pytest->-r requirements.txt (lin
e 8)) (1.1.1)
Requirement already satisfied: py>=1.8.2 in /Users/anubhavchaudhary/Desktop/cheapBuy/.venv/lib/python3.10/site-packages (from pytest->-r requirements.txt (lin
e 8)) (1.11.0)
Requirement already satisfied: MarkupSafe>=2.0 in /Users/anubhavchaudhary/Desktop/cheapBuy/.venv/lib/python3.10/site-packages (from Jinja2>=3.0->Flask->-r req
uirements.txt (line 4)) (2.0.1)
Requirement already satisfied: sniffio in /Users/anubhavchaudhary/Desktop/cheapBuy/.venv/lib/python3.10/site-packages (from trio~=0.17->selenium->-r requireme
nts.txt (line 2)) (1.2.0)
Requirement already satisfied: async-generator>=1.9 in /Users/anubhavchaudhary/Desktop/cheapBuy/.venv/lib/python3.10/site-packages (from trio~=0.17->selenium-
>-r requirements.txt (line 2)) (1.10)
Requirement already satisfied: outcome in /Users/anubhavchaudhary/Desktop/cheapBuy/.venv/lib/python3.10/site-packages (from trio~=0.17->selenium->-r requireme
nts.txt (line 2)) (1.1.0)
Requirement already satisfied: sortedcontainers in /Users/anubhavchaudhary/Desktop/cheapBuy/.venv/lib/python3.10/site-packages (from trio~=0.17->selenium->-r
requirements.txt (line 2)) (2.4.0)
Requirement already satisfied: wsproto>=0.14 in /Users/anubhavchaudhary/Desktop/cheapBuy/.venv/lib/python3.10/site-packages (from trio-websocket~=0.9->seleniu
m->-r requirements.txt (line 2)) (1.0.0)
Requirement already satisfied: cryptography>=1.3.4 in /Users/anubhavchaudhary/Desktop/cheapBuy/.venv/lib/python3.10/site-packages (from urllib3[secure]~=1.26-
>selenium->-r requirements.txt (line 2)) (35.0.0)
Requirement already satisfied: pyOpenSSL>=0.14 in /Users/anubhavchaudhary/Desktop/cheapBuy/.venv/lib/python3.10/site-packages (from urllib3[secure]~=1.26->sel
enium->-r requirements.txt (line 2)) (21.0.0)
Requirement already satisfied: cffi>=1.12 in /Users/anubhavchaudhary/Desktop/cheapBuy/.venv/lib/python3.10/site-packages (from cryptography>=1.3.4->urllib3[se
cure]~=1.26->selenium->-r requirements.txt (line 2)) (1.15.0)
Requirement already satisfied: pycparser in /Users/anubhavchaudhary/Desktop/cheapBuy/.venv/lib/python3.10/site-packages (from cffi>=1.12->cryptography>=1.3.4-
>urllib3[secure]~=1.26->selenium->-r requirements.txt (line 2)) (2.20)
Requirement already satisfied: h11<1,>=0.9.0 in /Users/anubhavchaudhary/Desktop/cheapBuy/.venv/lib/python3.10/site-packages (from wsproto>=0.14->trio-websocke
t~=0.9->selenium->-r requirements.txt (line 2)) (0.12.0)
Requirement already satisfied: colorama in /Users/anubhavchaudhary/Desktop/cheapBuy/.venv/lib/python3.10/site-packages (from crayons->webdriver-manager->-r re
quirements.txt (line 1)) (0.4.4)
Requirement already satisfied: pyparsing<3,>=2.0.2 in /Users/anubhavchaudhary/Desktop/cheapBuy/.venv/lib/python3.10/site-packages (from packaging->pytest->-r
requirements.txt (line 8)) (2.4.7)
WARNING: You are using pip version 21.2.3; however, version 21.3.1 is available.
You should consider upgrading via the '/Users/anubhavchaudhary/Desktop/cheapBuy/.venv/bin/python3 -m pip install --upgrade pip' command.
  ~/Doc/d/cheapBuy  on  dev_aakriti  python code/web/server_api.py                          ✓  cheapBuy   base   at 09:55:36 PM
 * Serving Flask app 'server_api' (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
 * Running on all addresses.
   WARNING: This is a development server. Do not use it in a production deployment.
 * Running on http://192.168.1.29:8080/ (Press CTRL+C to quit)
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 128-918-196
```

Original Repo Readme



README.md

🛠 Installation Steps

1. Clone the github repository at the preferable location in your system. You will need git to be preinstalled in the system. Once the repository is cloned in your system, with the help of cd command ,

```
git clone https://github.com/het-patel99/cheapBuy.git
cd cheapBuy
```

2. This project uses Python 3, so make sure that Python and Pip are preinstalled. All requirements of the project are listed in the `requirements.txt` file. Use pip to install all of those.

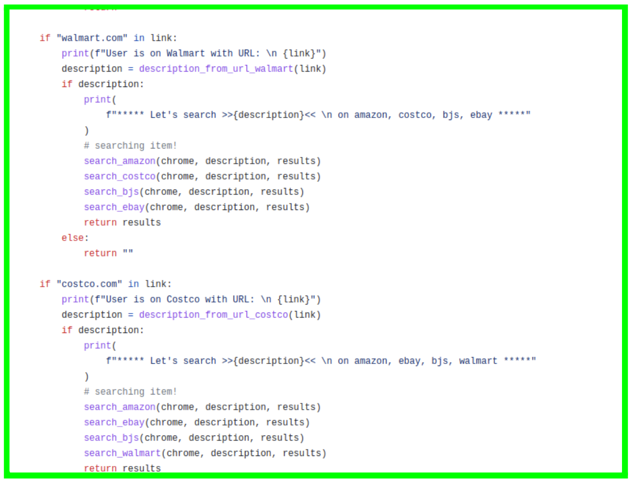
```
pip install -r requirements.txt
```

3. Check out the demo video to know about the use of the extension.

2. Redundant code:

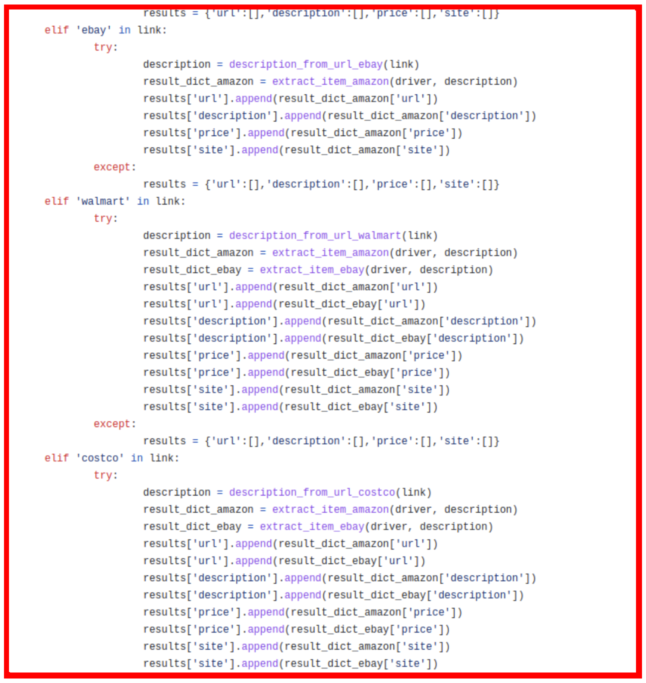
- in the web_scrapper.py the code was redundant for each scraper therefore a function is created to handle this and called whenever required in the program

Updated Code:



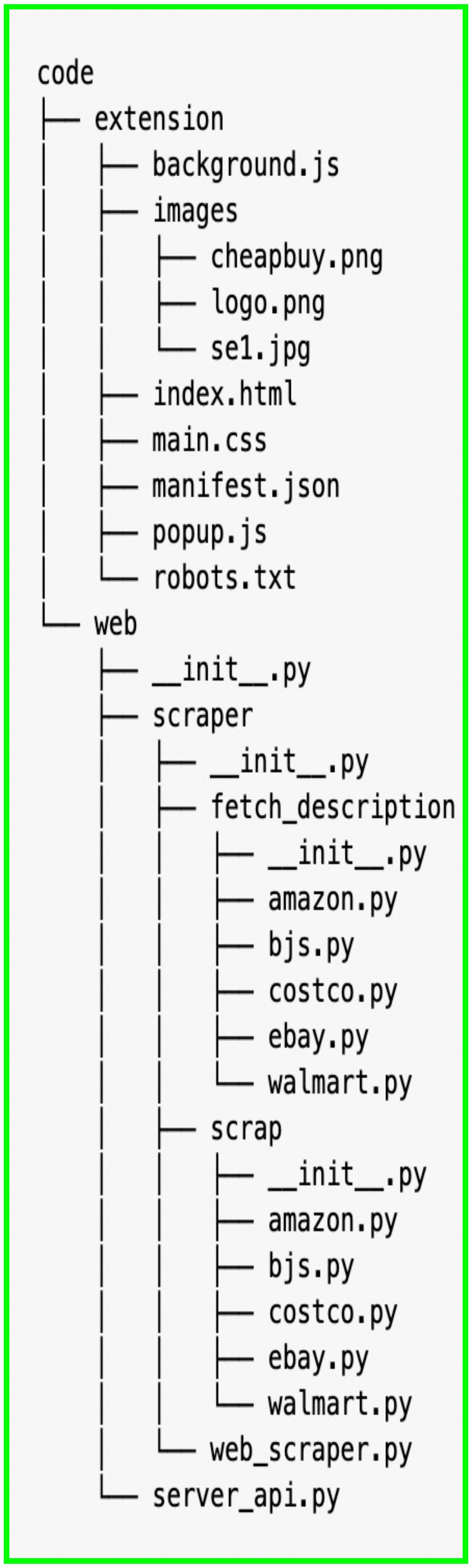
```
    if "walmart.com" in link:
        print(f"User is on Walmart with URL: \n {link}")
        description = description_from_url_walmart(link)
        if description:
            print(
                f"***** Let's search >>{description}<< \n on amazon, costco, bjs, ebay *****"
            )
            # searching item!
            search_amazon(chrome, description, results)
            search_costco(chrome, description, results)
            search_bjs(chrome, description, results)
            search_ebay(chrome, description, results)
            return results
        else:
            return ""

    if "costco.com" in link:
        print(f"User is on Costco with URL: \n {link}")
        description = description_from_url_costco(link)
        if description:
            print(
                f"***** Let's search >>{description}<< \n on amazon, ebay, bjs, walmart *****"
            )
            # searching item!
            search_amazon(chrome, description, results)
            search_ebay(chrome, description, results)
            search_bjs(chrome, description, results)
            search_walmart(chrome, description, results)
            return results
```

Code from original Repo:

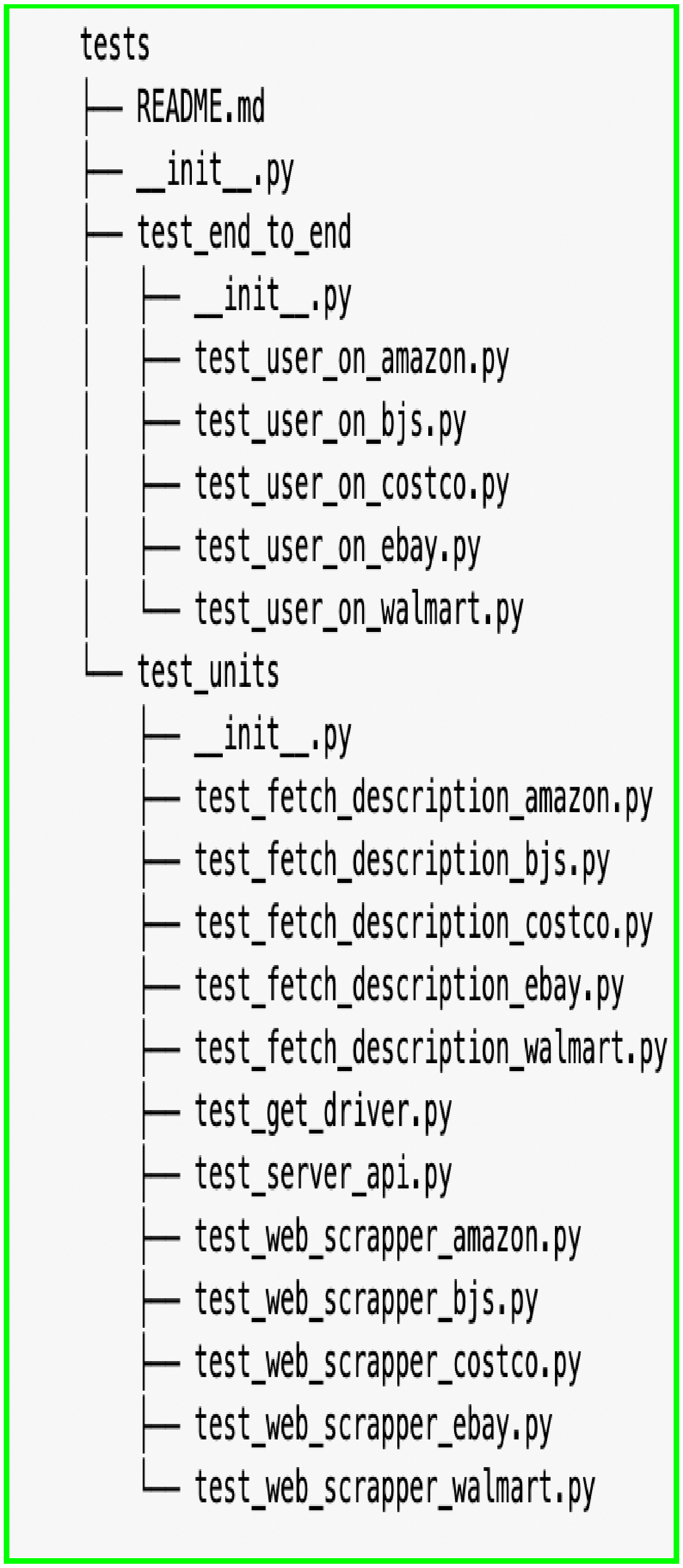


```
                results = {'url':[],'description':[],'price':[],'site':[]}
    elif 'ebay' in link:
            try:
                description = description_from_url_ebay(link)
                result_dict_amazon = extract_item_amazon(driver, description)
                results['url'].append(result_dict_amazon['url'])
                results['description'].append(result_dict_amazon['description'])
                results['price'].append(result_dict_amazon['price'])
                results['site'].append(result_dict_amazon['site'])
            except:
                results = {'url':[],'description':[],'price':[],'site':[]}
    elif 'walmart' in link:
            try:
                description = description_from_url_walmart(link)
                result_dict_amazon = extract_item_amazon(driver, description)
                result_dict_ebay = extract_item_ebay(driver, description)
                results['url'].append(result_dict_amazon['url'])
                results['url'].append(result_dict_ebay['url'])
                results['description'].append(result_dict_amazon['description'])
                results['description'].append(result_dict_ebay['description'])
                results['price'].append(result_dict_amazon['price'])
                results['price'].append(result_dict_ebay['price'])
                results['site'].append(result_dict_amazon['site'])
                results['site'].append(result_dict_ebay['site'])
            except:
                results = {'url':[],'description':[],'price':[],'site':[]}
    elif 'costco' in link:
            try:
                description = description_from_url_costco(link)
                result_dict_amazon = extract_item_amazon(driver, description)
                result_dict_ebay = extract_item_ebay(driver, description)
                results['url'].append(result_dict_amazon['url'])
                results['url'].append(result_dict_ebay['url'])
                results['description'].append(result_dict_amazon['description'])
                results['description'].append(result_dict_ebay['description'])
                results['price'].append(result_dict_amazon['price'])
                results['price'].append(result_dict_ebay['price'])
                results['site'].append(result_dict_amazon['site'])
                results['site'].append(result_dict_ebay['site'])
```

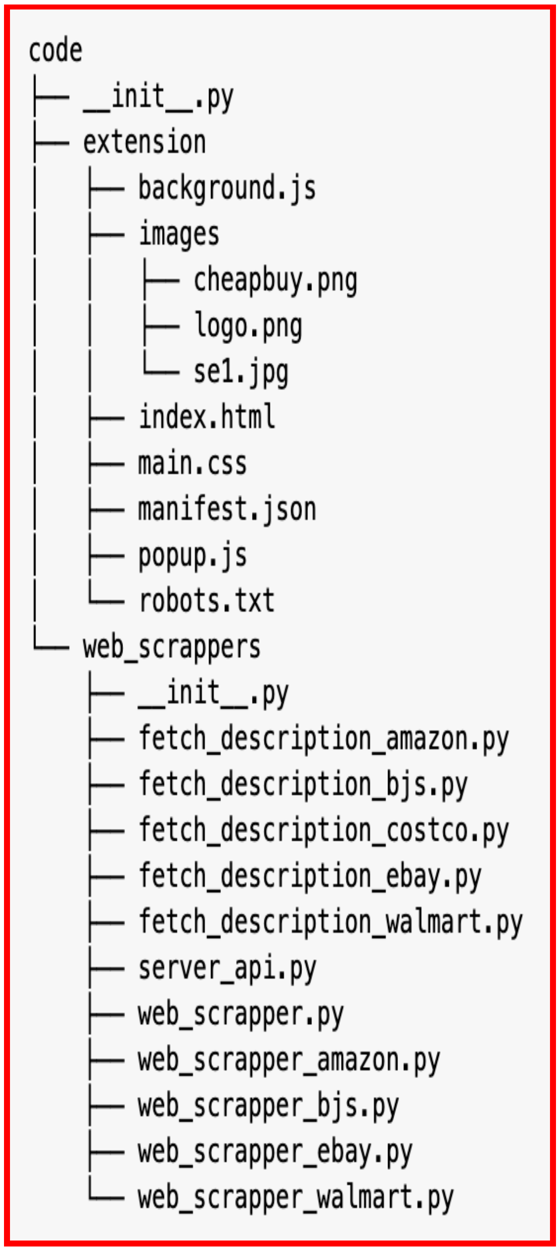
3. **Packaging issue:**
    - The source code is dumped all in one folder with no packaging done, the repo is refactored and the code is organized in a clear, easy read, and understandable structure.



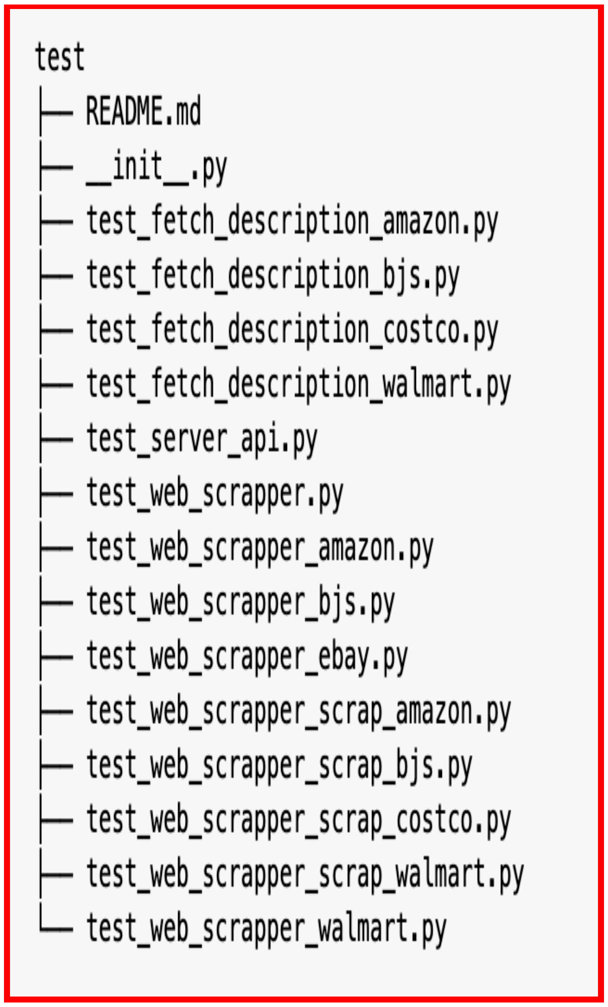
```
code
├── extension
│   ├── background.js
│   ├── images
│   │   ├── cheapbuy.png
│   │   ├── logo.png
│   │   └── se1.jpg
│   ├── index.html
│   ├── main.css
│   ├── manifest.json
│   ├── popup.js
│   └── robots.txt
└── web
    ├── __init__.py
    ├── scraper
    │   ├── __init__.py
    │   ├── fetch_description
    │   │   ├── __init__.py
    │   │   ├── amazon.py
    │   │   ├── bjs.py
    │   │   ├── costco.py
    │   │   ├── ebay.py
    │   │   └── walmart.py
    │   ├── scrap
    │   │   ├── __init__.py
    │   │   ├── amazon.py
    │   │   ├── bjs.py
    │   │   ├── costco.py
    │   │   ├── ebay.py
    │   │   └── walmart.py
    │   └── web_scraper.py
    └── server_api.py
```



```
tests
├── README.md
├── __init__.py
├── test_end_to_end
│   ├── __init__.py
│   ├── test_user_on_amazon.py
│   ├── test_user_on_bjs.py
│   ├── test_user_on_costco.py
│   ├── test_user_on_ebay.py
│   └── test_user_on_walmart.py
└── test_units
    ├── __init__.py
    ├── test_fetch_description_amazon.py
    ├── test_fetch_description_bjs.py
    ├── test_fetch_description_costco.py
    ├── test_fetch_description_ebay.py
    ├── test_fetch_description_walmart.py
    ├── test_get_driver.py
    ├── test_server_api.py
    ├── test_web_scrapper_amazon.py
    ├── test_web_scrapper_bjs.py
    ├── test_web_scrapper_costco.py
    ├── test_web_scrapper_ebay.py
    └── test_web_scrapper_walmart.py
```



```
code
├── __init__.py
├── extension
│   ├── background.js
│   ├── images
│   │   ├── cheapbuy.png
│   │   ├── logo.png
│   │   └── se1.jpg
│   ├── index.html
│   ├── main.css
│   ├── manifest.json
│   ├── popup.js
│   └── robots.txt
└── web_scrappers
    ├── __init__.py
    ├── fetch_description_amazon.py
    ├── fetch_description_bjs.py
    ├── fetch_description_costco.py
    ├── fetch_description_ebay.py
    ├── fetch_description_walmart.py
    ├── server_api.py
    ├── web_scrapper.py
    ├── web_scrapper_amazon.py
    ├── web_scrapper_bjs.py
    ├── web_scrapper_ebay.py
    └── web_scrapper_walmart.py
```



```
test
├── README.md
├── __init__.py
├── test_fetch_description_amazon.py
├── test_fetch_description_bjs.py
├── test_fetch_description_costco.py
├── test_fetch_description_walmart.py
├── test_server_api.py
├── test_web_scrapper.py
├── test_web_scrapper_amazon.py
├── test_web_scrapper_bjs.py
├── test_web_scrapper_ebay.py
├── test_web_scrapper_scrap_amazon.py
├── test_web_scrapper_scrap_bjs.py
├── test_web_scrapper_scrap_costco.py
├── test_web_scrapper_scrap_walmart.py
└── test_web_scrapper_walmart.py
```

4. **A lot of try-catch:**

   Initially, the code was flooded with a lot of try-catch blocks that were used without actually catching the error if needed.

5. **Edge case wasn't handled:** The old code was working for any other website other than [amazon, ebay, costco, bjs, walmart] such as office depot, which it should not. This edge was not handled instead for office depot the extension was going into the Bjs's part(else statement) and

scraping from respectively written other sites such as amazon, ebay. This was a Bug(code should handle it to notify/log on the server that the extension is not allowed to be used on the office depot website). Later it was fixed.

6. **Introduce relative imports:** The old code has each nested python module imported by adding to the system path. I introduced relative imports to the whole code base. This has improved code readability.

Feature Enhancement:

New feature enhancement - web scraper for BJ's and Costco are implemented Test cases are written to make sure all the functions and code is working fine

Code Understanding using Docstring:

Since, there were no comments found in the backend server code(python modules) or in the frontend extension code, it was quite hard to understand the end-to-end flow of the code. Therefore, The code is enhanced with docstrings attached to each and every functionality in the code along with test cases.