Name: Saianand Thevar

Internship Assignment

Aim: To scrape data from BigBasket and obtain product details like name, actual price and discounted price

Work done:

Initially the aim was attempted with the beautifulSoup library but that didn't work out as most e-commerce sites these days use a JS script in the web page response to fetch their product listing details which simple scraping like beautifulSoup can't really handle. Its more suited for static content.

As an alternative, the scrapy library was implemented with the integration of the pyppeteer library to successfully scrape the data we need. The pyppeteer library helps with the deploying and manipulation of headless browser (those without GUI) to fetch the web response and to run and load all the JS scripts which is done asynchronously and this needs to be kept in mind when implementing in your projects. For this particular task, the library used chrome browser as it was already installed but can be changed to other alternatives like chromium, firefox or phantomJS. Which gives the best performance needs to be tested as headless browsers are obviously more resource intensive.

Code:

Puppest.py

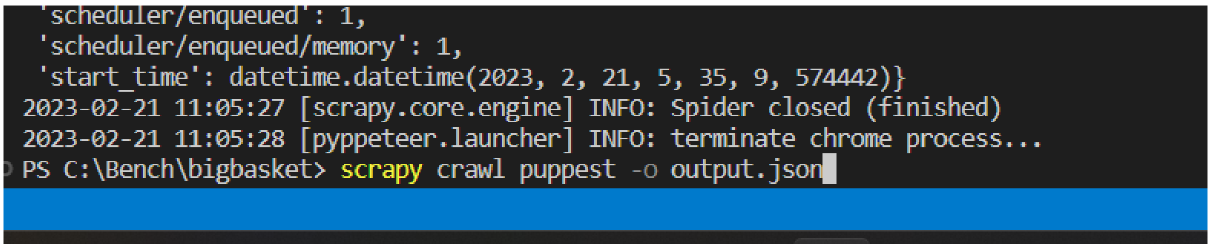
```python
import asyncio
import scrapy
from pyppeteer import launch


class PuppestSpider(scrapy.Spider):
    name = "puppest"
    allowed_domains = ["www.bigbasket.com"]
    start_urls = ["https://www.bigbasket.com/ps/?q=shampoo"]

    async def scrape(self, url):
        browser = await launch(headless=True)
        page = await browser.newPage()
        await page.goto(url)
        body = await page.content()
        await browser.close()
        return body

    async def parse(self, response):
        page_content = await self.scrape(response.url)
        selector = scrapy.Selector(text=page_content)
        for product in selector.css('div[qa="product"]'):
            product_name = product.css('div[qa="product_name"] a::text').get()
```

=

```
            actual_price = product.css('span.mp-price.ng-scope
span::text').get()
            discounted_price = product.css('span.discnt-price
span::text').get()
            yield {
                'product_name': product_name.strip() if product_name else
None,
                'actual_price': actual_price.strip() if actual_price else
None,
                'discounted_price': discounted_price.strip() if
discounted_price else None,
            }
```

Above is the primary code file which we’re using here, but its built with the help of the scrapy library and its method for a spider generation called genspider, executed from the CLI

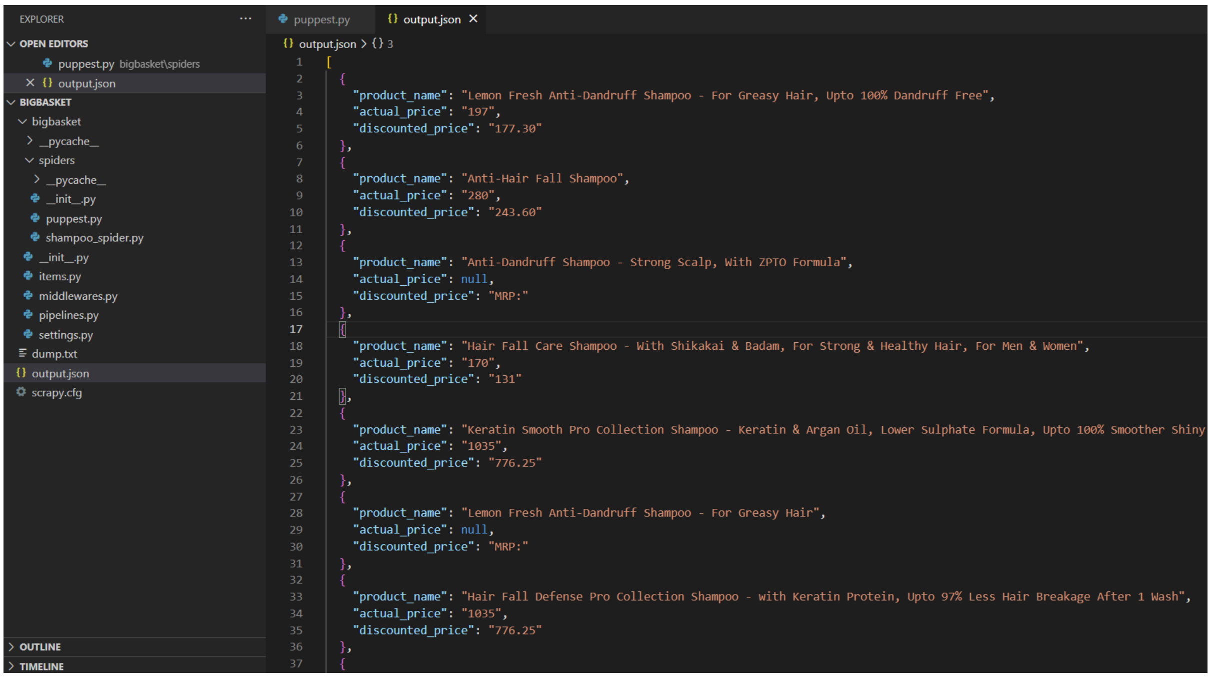
Results:



```
  'scheduler/enqueued': 1,
  'scheduler/enqueued/memory': 1,
  'start_time': datetime.datetime(2023, 2, 21, 5, 35, 9, 574442)}
 2023-02-21 11:05:27 [scrapy.core.engine] INFO: Spider closed (finished)
 2023-02-21 11:05:28 [pyppeteer.launcher] INFO: terminate chrome process...
 PS C:\Bench\bigbasket> scrapy crawl puppest -o output.json
```

Running the command as shown above runs your spider and saves the extracted data into output.json file which can be seen below

=



```
[
  {
    "product_name": "Lemon Fresh Anti-Dandruff Shampoo - For Greasy Hair, Upto 100% Dandruff Free",
    "actual_price": "197",
    "discounted_price": "177.30"
  },
  {
    "product_name": "Anti-Hair Fall Shampoo",
    "actual_price": "280",
    "discounted_price": "243.60"
  },
  {
    "product_name": "Anti-Dandruff Shampoo - Strong Scalp, With ZPTO Formula",
    "actual_price": null,
    "discounted_price": "MRP:"
  },
  {
    "product_name": "Hair Fall Care Shampoo - With Shikakai & Badam, For Strong & Healthy Hair, For Men & Women",
    "actual_price": "170",
    "discounted_price": "131"
  },
  {
    "product_name": "Keratin Smooth Pro Collection Shampoo - Keratin & Argan Oil, Lower Sulphate Formula, Upto 100% Smoother Shiny
    "actual_price": "1035",
    "discounted_price": "776.25"
  },
  {
    "product_name": "Lemon Fresh Anti-Dandruff Shampoo - For Greasy Hair",
    "actual_price": null,
    "discounted_price": "MRP:"
  },
  {
    "product_name": "Hair Fall Defense Pro Collection Shampoo - with Keratin Protein, Upto 97% Less Hair Breakage After 1 Wash",
    "actual_price": "1035",
    "discounted_price": "776.25"
  },
  {
```

Conclusion:

So far the data is extractable from e-commerce site but further tests need to be conducted for different sites under different situations, along with the options to run this on an AWS EC2 instance and also for handling pagination of websites.