Software Engineer Assignment

Credentials to connect to Mongo
User: Cliff
Password: G4gxceAlfN6NN51Q

Task 1:

- Used scrapy shell to scrap footwear link.
- Obtained xpath to the div which contains all the products
- Similarly, with some experiments I could obtain the xpath to the asked attributes of interest.
- Similarly, the second link for footwear was scrapped and notices that the structure for both the pages is the same.
- Created scrapy project which created the folder structure containing
- Inside items.py
    - Defined class `CliffItem` to define structure of data.
- Inside cliff/spiders
    - Created class `CliffSpider`, a spider named "`cliffSpider`" and defined base url along with logic to scrap both the links.
    - Spider would go to the site mentioned and create item and yield it.
- Faced some errors:
    - In settings.py, `ROBOTSTXT_OBEY = False` was changed from True to False to allow scraping
- I tested the code till now and saved the output in a json file.
- Next Step was to build a pipeline that would automate the whole process without writing the scrapped records into a json file, instead writing directly to mongoDB.
- Inside pipelines.py
    - Defined `MongoDBPipeline` with connection string to connect to mongoDB and defined default parse_item function.
    - When spider crawls, it collects data in item object and send it to pipeline.
    - Here the init method is called by default and then process_item() is called.
    - Init method establishes connection to the mentioned db and collection.
    - Process_item write item received to db and reference is passed to spider to send next item to write to db.
- In Settings.py
    - `ITEM_PIPELINES = {'cliff.pipelines.MongoDBPipeline':0}`
    - Register pipeline class in settings.py

Task 2:

- Connected with mongoDB using mongosh.
- Ran queries to answer the asked questions.
- Most Interesting: How many products have discount % greater than 30%?

- Defined divide operation separately and used with less than operator to get discounted products.
    - let cmp = { $divide: ["$sale_price", "$original_price"] }
    - db.flipkart.find({$expr: {$lt : [cmp, NumberDecimal("0.7")]}}).count()