**Q. Use scrapy to crawl any of the E-commerce website of your choice.**

**The following information needs to be extracted from the page for the corresponding product:**

1. **Product Name**
2. **Product Price**
3. **Product Discount**
4. **Product Image (URL)**

**PROCEDURE:**

1. **SETUP**
2. Installing scrapy library onto the device or virtual environment.

Enter in the terminal:

pip install scrapy

1. Move to the folder where you want to create your scrapy project in the terminal and enter:

scrapy startproject ecomscrape

to create a scrapy project with the name ecomscrape at the location

1. Move up to the root directory of the scrapy project using terminal command:

cd ecomscrape

1. In the ecomscrape directory of the root scrapy directory, another directory named spiders will be present where we have to create the spider we want to use.

Create a python3 file in the spiders directory with a name of your choice. (ecomSpider.py here)

1. **PROGRAMMING**
2. Open the ecomSpider.py file in an IDE.
3. First we want to import the scrapy library using:

import scrapy

1. Create a class using the scrapy library with scrapy.Spider class

class PostsSpider(scrapy.Spider):

1. Inside the class, assign the spider we are creating an unique name as an identifier to call the spider later using terminal as:

name=”posts”

1. Now, specify the list of base urls from where you want the scraper to start crawling as:

start\_urls=[“https://www.flipkart.com/search?q=earphones”]

1. Now, create a ‘parse’ function inside the class. The parse function is the default function which is called when the spider is run.
2. Type your program for data extraction inside the parse function. (use scrapy shell ‘URL’ command in terminal and test your snippets of logic in there to check if any particular query works or not)
3. Run the program to compile it for any errors. (This is not execution of the spider, its just compilation of the program to save it and check it for errors)
4. **EXECUTION**
5. If the robots.txt file of the website disallow the domain you want to access then in the settings.py file of the scrapy project set ROBOTSTXT\_OBEY Boolean value to false and proceed. (It might be against the sites policy so check for it beforehand)
6. In the terminal enter the command:

scrapy crawl posts -o *filename*.json

1. The json file will be created with the required data in the root directory of the scrapy project.

**CODE:**

import scrapy

class PostsSpider(scrapy.Spider):

    name= "posts"

    start\_urls=["https://www.flipkart.com/search?q=earphones"]

    def parse(self, response):

        for p in response.css('div.\_4ddWXP'):

            yield{

                'Name': p.css('a.s1Q9rs::text').get(),

                'Price': p.css('a.\_8VNy32 div.\_25b18c div.\_30jeq3::text').get().split('\u20b9')[1],

                'Images': p.css('img::attr(src)')[0].get(),

                'Discount':p.css('div.\_3Ay6Sb span::text').get().split(' ')[0]

            }

**OUTPUT:**







