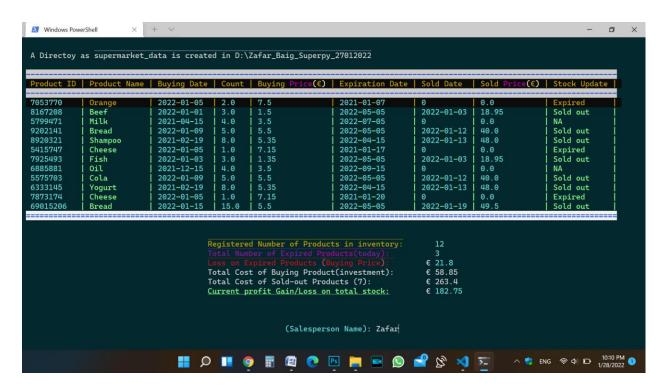
## **User's Manual Superpy**

Command Line Interfaces (CLI) are programmatic scripts and libraries that are executed with a unique purpose; mostly these are utilized or operate with the help of command line or from a Windows Power Shell and are control in most of the operating system.

There are so many Python libraries and modules that help to build a command line app from parsing arguments and options to flagging to full blown CLI "frameworks" which do things like colorized output, tabular data, progress bars, sending email and so on.

In the following Supermarket SuperPy app a number of different modules are imported to run to solve the problems that are described in the assignment. First of all to run this program we must to execute it with main file; c: python3 main.py (Table.1).

Consequently, this will create a directory known as "supermarket\_data". It encloses all the required inventory files that have been converted with an extension of .csv. When a new item (product) is going to buy or sell, all the relevant information in these newly formed files are stored for further processing.



**Table 1. Main Execution of Program** 

There are four different commands used to acquire proper result of the problems described in the assignment. These CLI explained and simplified as an easy way of conducting program for the users. To get help for the said program <a href="mailto:py-help">python3</a> main.py <a href="help">help</a> (Table.2) is use to follow instruction given to user.

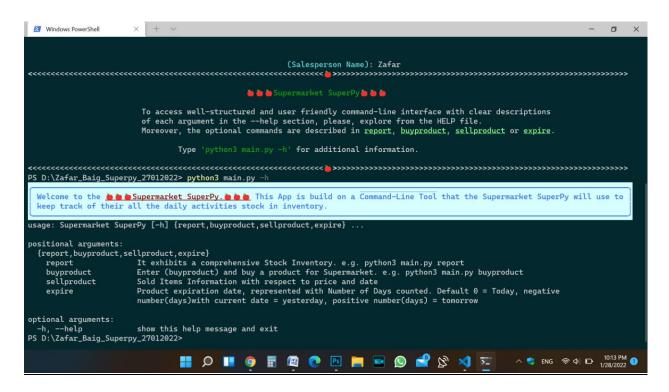


Table 2. Help

### 1. Buy Product.

To buy a product for the Supermarket a process of item registration is introduced. It will use to keep track of their all the daily activities stock in inventory. Consequently a number of questions require for the registration of buying a product are answer by entering them.

Moreover it makes sure the right product is going to be entering manually, with relevant attributes e.g. Product Name, Buying Date, Price, Expiration Date etc. etc. In this way a command line tool helps to store all the attributes of a product that is going to be bought for Superpy inventory.

After launching the program command line script (<u>python3 main.py buyproduct</u>) is contacted with a number of input fields about the attributes of a product; (Table 3).

Product Name: Apple

Date of Buying: 2022-01-27

• Product Amount: 4

Buying Price: 2.75

Expiration Date: 2022-08-30

After completion of data entries in the different input fields, supermarket database is updated with all the values that are stored in inventory with their unique product-id. (Table 4) The unique ID numbering of a product plays a vital role in the database. This will help to avoid duplication of the same product's name with different Product ID numbers.

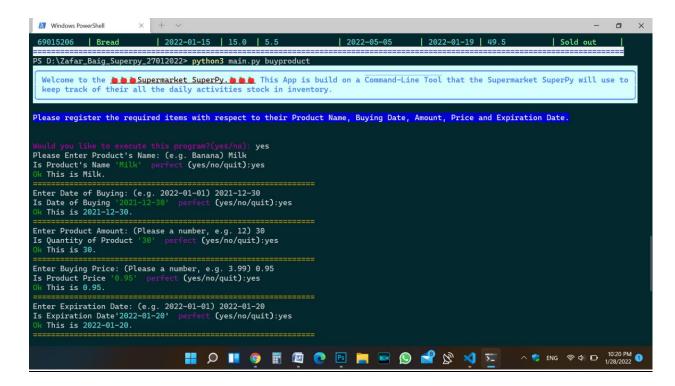


Table 3. Buying a product with entering different attributes.

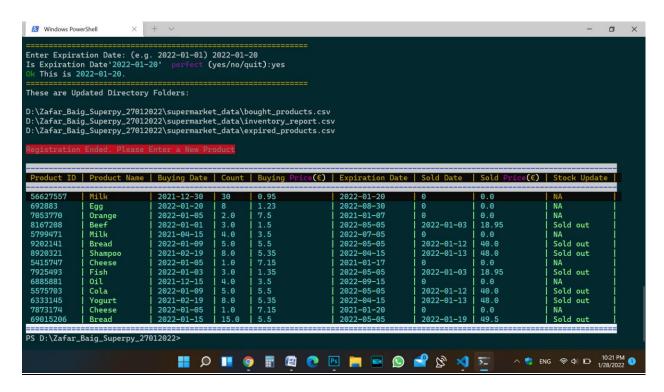


Table 4. Bought product in inventory

#### 2.Sell Product:

One of the purposes of introducing this app is, to hold the data of selling product stored in inventory. When certain product is going to sell another command line "sellproduct" with argument (-sp -- soldprice) is used with this script. e.g. <a href="python3 main.py -sellproduct - id 345636 - sp 35.70">python3 main.py - sellproduct - id 345636 - sp 35.70</a> (Table 5 & 6). In product specification, when a product is sold out it will clearly show the product ID, date of sold and sold price.

Besides, sellproduct command line can also compute a summary of sold products with total price of sold item. If a product is not in the inventory it also shows the product does not exist, the product is expired or sold out.

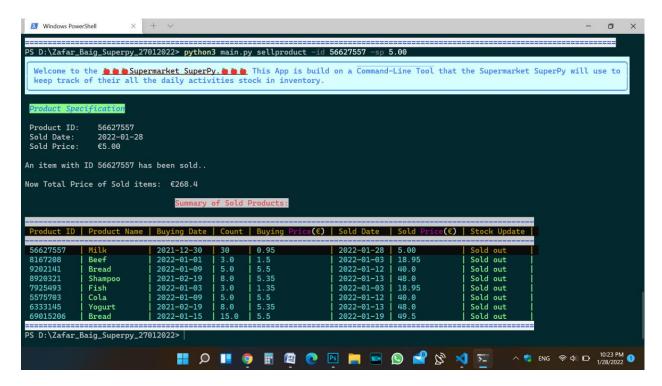


Table 5. Selling a product and Summary of Sold items.

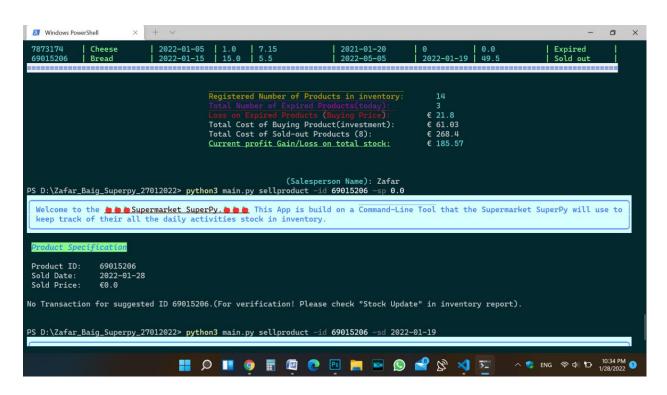


Table 6. Existed sold item in inventory.

# **3.Expire Product**

This set out a comparison of different schedules in stipulated dates with the expiration of products stored in database. This can be done by computing the number of days passed or advance numbers of days since then and simply set by the command.

The query is simply performed by entering negative number of days (e.g. <u>python3 main.py expire -nd - 15</u>) which print-out inventory report of the past days The default is zero number of days indicates inventory report publish for today. (Tables 7, 8 & 9),

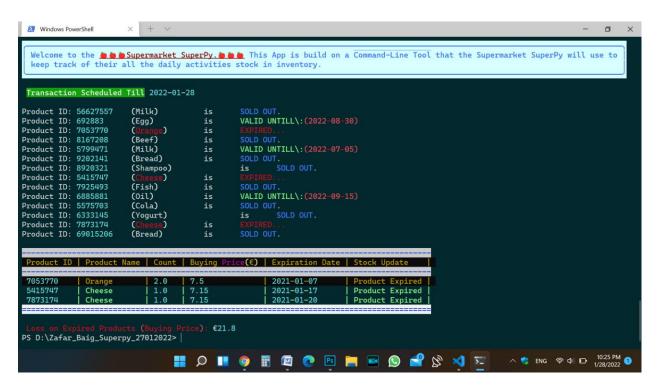


Table 7. Expired products for today

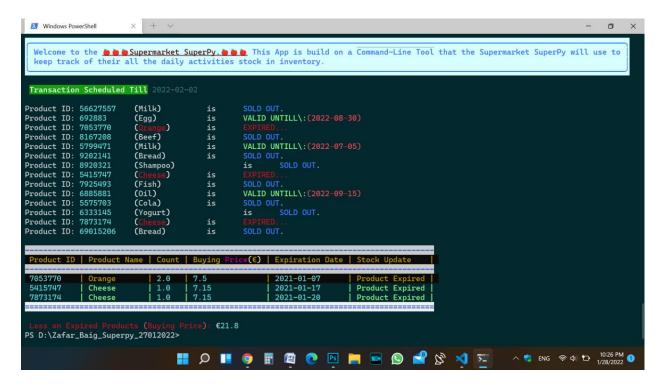


Table 8. Expired products exhibit advanced number of days

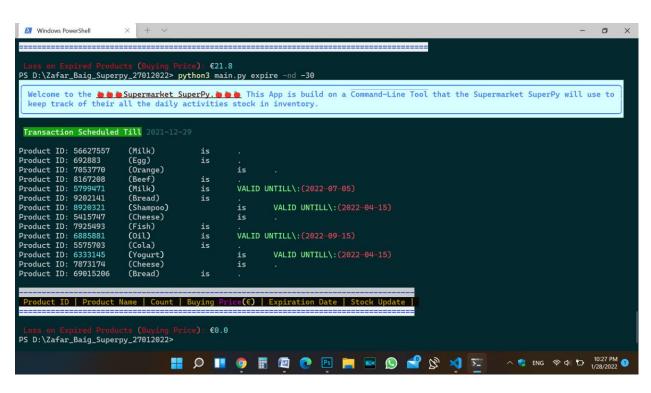


Table 9. Expired products for the past 30 numbers of days.

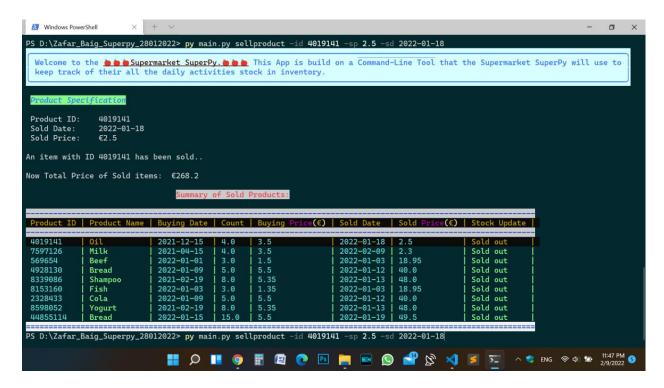


Table 10. Sell product with name, price and specified date.

# 4. Report Inventory.

The command line "report" makes a comprehensive inventory report of sold products, bought products and status of product expired. E.g. <u>python3 main.py report</u>.

Furthermore report also exhibits the products with respect to Registered Number of items, Total Costs, Total Number of sales, Total Number of Expired Items and Current Profit Gain/Loss on Total Stock(Table 10).

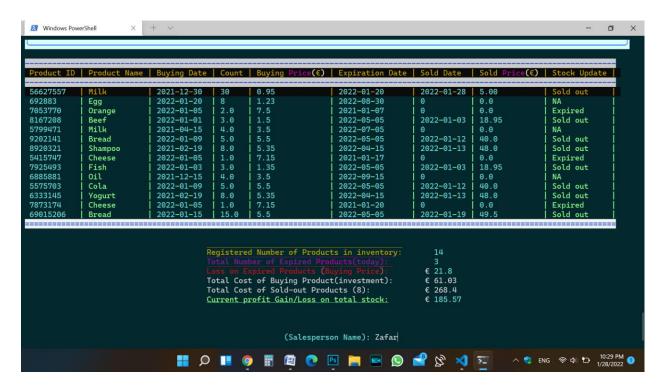


Table 11. Inventory Report showing Current profit on total stock.

## Supermarket SuperPy Guidelines

{buyproduct: Buy a product with different attributes.

Sellproduct: Sell a product with different attributes.

Expired: To inquire condition of a product in inventory with stipulated dates.

Report: A comprehensive report of all the stock in inventory with gain

Profit/Loss}