# ASSIGNMENT 3

LAVANYA SASIKALA

156621211

**Program Requirements:**

**Purpose:**

The purpose of this program is to implement a store stock calculator for a small business.

**Scope:**

The program allows the user to open a CSV file and display the contents of the file in a pretty table format. The CSV file contains fields like Item, Current Stock, Price per Item, and Previous Sales. The pretty table displayed after opening the file contains fields like Sl. No, Item, Current Stock, and Price per Item. The format of the table is as below.

# Item Current Stock Price per Item

==================================================

1. Apple 3 $ 1.00

2. Banana 4 $ 2.50

3. Orange 5 $ 1.50

**Table1**

During the day, the user will be able to enter items that have been sold using a prompt to select the Sl. No. of the item or pressing ‘e’ to indicate the end of the day. Each time the user enters a number, the current stock of the corresponding item will be decremented by 1. If the current stock reaches 0, stop decrementing, instead increment the demand for that item. If ‘e’ is selected, then three reports need to be generated.

1. **Total Sales**: A Table that indicates the total sales, including the totals for each item and a grand total as below:

# Item Sales $ / Item Total

===============================================

1. Apple 3 $ 1.00 $ 3.00

2. Banana 3 $ 2.50 $ 7.50

3. Orange 2 $ 1.50 $ 3.00

TOTAL $ 13.50

**Table2**

1. **Sales Trends:** This tableinterprets the “Previous Sale” field in the imported CSV file and compares the sale made on a day with that of the previous day. A comparison of total sales versus that of the previous day is made to calculate the trend. For example, if a day’s sale was double that of the previous day, then the trend is 100.00%.

The trend is calculatedas:

((Today’s Sale-Previous day sale)/Prev. day sale) \* 100

The table should be as below:

# Item Sales Previous Sales Trend

=============================================================

1. Apple 3 4 -25.00%

2. Banana 3 2 50.00%

3. Orange 2 5 -60.00%

TOTAL SALES TODAY 8

TOTAL SALES YESTERDAY 11

TREND -27.27%

**Table 3**

1. **Restock Report:** This report tells the manager how many of each item will be needed to address average demand. The average demand is calculated by getting the average for today and previous sales, and then rounding to the closest integer. The table should be as below:

# Item Average Demand Current Stock From Warehouse

==========================================================

1. Apple 4 0 4

2. Banana 2 1 1

3. Orange 4 3 1

**Table 4**

**Finally,** a CSV file with the current updated stock needs to be generated, which can be imported for the next day’s activity.

**Required Features and Implementation:**

1. Import the CSV file and read it to a list of dictionaries and print it with the required fields in a pretty table format as in **Table 1.**
2. The user needs to enter the serial number of the corresponding item for which a sale was made.
3. Each time the user selects a number, the stock of the item against the number needs to be decremented by 1. If the stock reaches 0, stop decrementing.
4. When the user presses ‘e’ to indicate the end of sale for the day, the following reports need to be generated.
5. Total Sales as in Table 2
6. Sales Trend as in Table 3
7. Restock report as in Table 4

**Valid questions to ask before implementation.**

1. Is the list of items in the initial CSV file static; or needs to be updated dynamically?
2. Is it required to change the price of the items dynamically?
3. What if the user enters a number not valid while entering the sale item?
4. What if the user enters a character other than ‘e’ to mark the end of the day?

**Data types:**

1. List: The input CSV file is read as a list of dictionaries.
2. Dictionary: Each line (except the first line) of the CSV file is converted to a dictionary.
3. Int: For the Number of stock and demand
4. Float: For the price per item and totals.
5. Strings: For the item name

**What needs to be saved finally?**

At the end of the day, 3 tables (Table 2, Table 3, and Table 4) need to be generated. Also, a CSV file needs to be created with the updated stock.

**What can go wrong?**

1. User enters a number that is not valid.
2. User enters a character other than ‘e’ to mark the end of the day.

**What different checks need to be implemented?**

1. Check for entering a valid number for the sale item.
2. Check for entering the character ‘e’ to mark the end of the day.
3. Check for the condition when the stock of an item reaches 0.