Assignment SuperPy Winc

SuperPy is a command-line tool that can keep track of Supermarket inventory. The core functionality of the program is about keeping track and producing reports on the various kinds of data.

import sys

import csv

from texttable import Texttable

from datetime import timedelta, datetime

The imports mentioned about are used in the project regarding the following functionalities:

1. Sys: This import is used to get command line arguments as well as stdout and stdin purposes
2. Csv: This import is used in reading csv file and writing to the csv file
3. Texttable: This is an external module to show the data in a table view for an interactive view.
4. Datatime: Import to work on the date and timedelta in this particular project is used to increase the date or decrease in the date. Like –advance-time and –yesterday statements.

def get\_inventory():

   '''

      Reads the bought.csv file row by row and return

      the grand list of all the rows

   '''

   with open('bought.csv', mode='r') as csv\_file:

    csv\_reader = csv.reader(csv\_file, delimiter=',')

    all\_rows = []

    for row in csv\_reader:

       all\_rows.append(row)

    return all\_rows

The function mentioned above is used to read the csv file. It reads the elements of the csv file line by line and keeps it appending to the grand list. Then that grand list with all the inventory items is returned. This makes it easier to understand, and get the item on the basis of the date as well.

def validate\_date(date\_string):

   '''

      This function accepts the input string (date) and check whether the date is valid or not

      usually it is used in the command like buy product or sell product

   '''

   try:

      datetime.strptime(date\_string, '%Y-%m-%d')

      return True

   except ValueError:

      sys.stdout.write("Invalid expiration date. It must be formatted like YYYY-MM-DD")

      return False

I used this function to validate the date. For example if the user tries to sell or buy the item and enters the invalid date, it will print the error and shows the user an example of the valid format.

def get\_date():

   '''

      Function that reads the text file where the date is stored.

      The date is stored in txt file becuase we have to advance the time

      some times.

   '''

   file\_contents = open("current\_date.txt", "r").read()

   return file\_contents

This function basically reads the current-\_date.txt file and returns it contents. Which is basically the current date the user is working on. When a user advances the time this date is modified in the text file and hence the current date becomes the advanced date.

def buy\_product(id, product\_name, price, expiry, buy\_date):

   '''

      Takes product name, price, expiry and buy date as arguments

      and open bought.csv file. Then it writes the data to the csv file

   '''

   with open('bought.csv', mode='a', newline='') as file:

       writer = csv.writer(file, delimiter=',', quotechar='"', quoting=csv.QUOTE\_MINIMAL)

       writer.writerow([id, product\_name, buy\_date, price, expiry])

   sys.stdout.write("OK")

This function simply takes the arguments and write the contents to the bought.csv file.

def sell\_product(id, product\_name, product\_price, inventory, current\_date):

   '''

      Initializes the product to None, that;s because if the product

      is not found it will display an error.

      If the product is found it will open the sold.csv file and write the information

      to that file.

   '''

   product = None

   for item in inventory:

      if item[1] == product\_name:

         product = item

   if product == None:

      sys.stdout.write("ERROR: Product not in stock")

   else:

      with open('sold.csv', mode='a', newline='') as file:

       writer = csv.writer(file, delimiter=',', quotechar='"', quoting=csv.QUOTE\_MINIMAL)

       writer.writerow([id, product[0], current\_date, product\_price])

       sys.stdout.write("OK")

It takes the arguments, in which the product name is the important argument. It checks each item if the product name matches. If the product name matches then it writes the contents to the sell.csv otherwise it shows the error message.

def report\_inventory(current\_date, inventory, day):

   inventory\_to\_display = []

   inventory\_to\_display.append(inventory[0])

   if day == "--yesterday":

      dt = datetime.strptime(current\_date, '%Y-%m-%d')

      yesterday = dt - timedelta(days = 1)

      current\_date = yesterday.strftime('%Y-%m-%d')

   for item in inventory:

      if item[2] == current\_date:

         inventory\_to\_display.append(item)

   t = Texttable()

   t.add\_rows(inventory\_to\_display)

   sys.stdout.write(t.draw())

This function prints the inventory on the basis of the day selected. So if the value of day is yesterday, it decreases the day and displays the data accordingly. Otherwise it displays the data on the basis of current date read from current\_date.txt file

def advance\_time(days\_to\_increase):

   current\_date = get\_date()

   dt = datetime.strptime(current\_date, '%Y-%m-%d')

   yesterday = dt + timedelta(days = days\_to\_increase)

   current\_date = yesterday.strftime('%Y-%m-%d')

   file = open("current\_date.txt", "w")

   file.write(current\_date)

   file.close()

   sys.stdout.write("OK")

Reads the current\_date.txt file and increases the date based on days entered by the user in the command.

def report\_revenue(report\_date, current\_date):

   revenue = 0

   if report\_date[0] == "--yesterday":

      current\_date = get\_date()

      dt = datetime.strptime(current\_date, '%Y-%m-%d')

      yesterday = dt + timedelta(days = days\_to\_increase)

      current\_date = yesterday.strftime('%Y-%m-%d')

   elif report\_date[0] == "--date":

      current\_date = report\_date[1]

   with open('sold.csv', mode='r') as csv\_file:

    csv\_reader = csv.reader(csv\_file, delimiter=',')

    for row in csv\_reader:

       if row[2] == current\_date:

         revenue += float(row[-1])

   sys.stdout.write("Revenue so far: " + str(revenue))

This function reads the csv file and check if the current date is equal to the date requested. If the date matches it calculated the revenue and prints the information.

def main():

   #Holds current date frin the text file

   current\_date = get\_date()

   #Holds the inventory

   inventory = get\_inventory()

   try:

      last\_item\_id = int(inventory[-1][0])

   except ValueError:

      last\_item\_id = 1

   args = sys.argv

   if args[1] == "buy":

      product\_name = args[3]

      price = args[5]

      expiry = args[7]

      buy\_date = current\_date

      if validate\_date(expiry):

         buy\_product(last\_item\_id + 1, product\_name, price, expiry, buy\_date)

   elif args[1] == "report":

      if args[2] == "inventory":

         report\_inventory(current\_date, inventory, args[3])

      elif args[2] == "revenue":

         if args[3] == "--date":

            report\_revenue([args[3], args[4]], current\_date)

         else:

            report\_revenue([args[3]], current\_date)

   elif args[1] == "sell":

      product\_name = args[3]

      product\_price = args[5]

      sell\_product(last\_item\_id, product\_name, product\_price, inventory, current\_date)

   elif args[1] == "--advance-time":

      advance\_time(int(args[2]))

This simple is the main function which is used to call the other functions, on the basis of user inp