```
import os,getpass, shutil, zipfile, random

#declarations
ZIP_ARCHIVES = 'Complaints\\Zip Archives'
FILE_ARCHIVES = 'Complaints\\File Archives'
PROCESSING = 'Complaints\\Processing'
MAIN_DIRECTORY = 'Complaints'
MASTER_FILE = os.path.join(MAIN_DIRECTORY, 'Complaints_Master.csv')

#move to the user desktop
the_desktop = os.path.join('C:\\Users',getpass.getuser(),'Desktop')
os.chdir(the_desktop)

def setup():
    #make or remake the main folder structure
    if os.path.isdir(MAIN_DIRECTORY):
        shutil.rmtree(MAIN_DIRECTORY)
        os.makedirs(FILE_ARCHIVES)
        os.makedirs(ZIP_ARCHIVES)
        os.makedirs(PROCESSING)
    else:
        os.makedirs(ZIP_ARCHIVES)
        os.makedirs(FILE_ARCHIVES)
        os.makedirs(PROCESSING)

    #set up master file with header
    full_path = os.path.join(MAIN_DIRECTORY, 'Complaints_Master.csv')
    if not os.path.isfile(full_path): #.isfile won't work with path.join
        with open (os.path.join(MAIN_DIRECTORY,
'Complaints_Master.csv'),'a') as master_csv:
            master_csv.write('Complaint ID, Date Received, Company, Product,
Issue\n')

def unpacking():
    #fill active_zip_files_list with names of zipped complaint files
    active_zip_files_list=[]

    #create a list of all zip files to process
    desktop_files = (os.listdir())
    for i in desktop_files:
        if i.startswith('aggregate_complaints_'):
            active_zip_files_list.append(i)
            full_path = os.path.join(ZIP_ARCHIVES,i)
            #SHUTIL.MOVE normal, SHUTIL.COPY to debug
            shutil.move(i,full_path)
    os.system('cls') #show them this
    print('Please wait, processing ' + str(len(active_zip_files_list)) + '
zipped complaint file(s)')

    #this loads non-duplicate files into the processing folder
    for i in active_zip_files_list:
```

```
        full_path = os.path.join(ZIP_ARCHIVES,i)
        active_zip = zipfile.ZipFile(full_path)
        active_file_list = active_zip.namelist()
        active_zip.close()
        archive_file_list = os.listdir(FILE_ARCHIVES)
        for j in active_file_list:
            if j not in archive_file_list:
                exampleZip = zipfile.ZipFile(full_path)
                exampleZip.extract(j, path= PROCESSING)
                exampleZip.close

    #processing one file at a time
    f2p = os.listdir(PROCESSING)
    for i in f2p:
        full_path = os.path.join(PROCESSING, i)
        archive_path = os.path.join(FILE_ARCHIVES, i)
        with open(full_path,'r') as json_file:
            next(json_file)
            for line in json_file:
                if len(line) < 3: #avoid the last line
                    break
                line_split = line.split(":")

                #get complaint id
                #do not assume the id length
                #have to approach everything after comments from the right
                cid = line_split[-1].strip()
                cid = cid.strip('"},')

                #get date_received
                date_received = line_split[1]
                date_received = date_received.split('"')
                date_received = date_received[1].strip()

                #get company
                #have to approach everything after comments from the right
                company = line_split[-11]
                company = company.split('"')
                company = company[1]
                company = company.replace(',',' ') #remove the commas
                company = company.replace('  ',' ')
                company = company.strip()

                #get product
                product = line_split[2]
                product = product.split('"')
                product = product[1].strip()
                product = product.replace(', ',' ') #remove the commas

                #get issue
                issue = line_split[4]
```

```
                issue = issue.split('"')
                issue = issue[1].strip()
                issue = issue.replace(',',' ')
                issue = issue.replace('  ',' ')

                with open (os.path.join(MAIN_DIRECTORY,
'Complaints_Master.csv'),'a') as master_csv:
                    master_csv.write(cid + ','
                                + date_received + ','
                                + company + ','
                                + product + ','
                                + issue + '\n')

        #move the processed file
        shutil.move(full_path,archive_path)

    os.system('cls') #show them this
    print('Complaint file processing is complete\n')
    input('Press Enter to continue')
    os.system('cls') #show them this

def cleanup():

    #initialize lists
    cid_list = []
    no_dupes_list = []

    #read all records from csv into master_csv list
    with open (os.path.join(MAIN_DIRECTORY, 'Complaints_Master.csv'),'r') as
master_csv:
        next(master_csv)
        for line in master_csv:
            cid_list.append(line)

    os.system('cls') #show them this
    print('Removing duplicate records, please wait\n')

    #make list of all records with no duplicates
    print('Number of current records: '.rjust(45) + str(len(cid_list)))
    for i in cid_list:
        if i not in no_dupes_list:
            no_dupes_list.append(i)

    #remove csv, make csv fresh from no dupes list
    os.remove(os.path.join(MAIN_DIRECTORY, 'Complaints_Master.csv'))
    with open (os.path.join(MAIN_DIRECTORY, 'Complaints_Master.csv'),'a') as
master_csv:
        master_csv.write('Complaint ID, Date Received, Company, Product,
Issue\n')
        for i in no_dupes_list:
            master_csv.write(i)
```

```
147
148     #reporting some stats
149     print('Number of records after removing duplicates: ' +
…  str(len(no_dupes_list)))
150     removed = len(cid_list)-len(no_dupes_list)
151     print()
152     print('Duplicate records removed: '.rjust(45) + str(removed))
153     input('\n\nPress Enter to continue')
154
155 def report():
156
157     #initialize list and load all records into csv_loaded list
158     csv_loaded=[]
159     with open (os.path.join(MAIN_DIRECTORY, 'Complaints_Master.csv'),'r') as
…  master_csv:
160         next(master_csv)
161         for line in master_csv:
162             csv_loaded.append(line)
163
164     #initialize list and load all records into product_list
165     product_list=[]
166     for i in csv_loaded:
167         i = i.split(',')
168         i=i[3]
169         if i not in product_list:
170             product_list.append(i)
171
172     while True:
173         issue_list=[]
174         company_list=[]
175         record_list=[]
176         os.system('cls') #show them this
177         print('AVAILABLE PRODUCTS')
178         print('------------------\n')
179         product_list.sort()
180         for i in range(1,len(product_list)+1):
181             print(str(i).rjust(2) + ' ' + product_list[i-1]) #right justify
…  line numbers
182         print()
183         choice = input('Enter the product number (zero to exit): ')
184         try:
185             int(choice)
186         except:
187             os.system('cls') #show them this
188             print('That is not a valid product number\n')
189             input('Press enter to continue')
190             os.system('cls') #show them this
191             continue
192         if int(choice) == 0:
193             break
194         choices_list = list(range(1,len(product_list)+1))
```

```
        if int(choice) not in choices_list:
            os.system('cls') #show them this
            print('That is not a valid product number\n')
            input('Press enter to continue')
            os.system('cls') #show them this
            continue

        choice_text = (product_list[int(choice)-1])

        for i in csv_loaded:
            myline = i.split(',')
            product = myline[3]
            if product.startswith(choice_text):
                record_list.append(myline[0])
                issue = myline[4][:-1]
                if issue not in issue_list:
                    issue_list.append(issue)
                company = myline[2]
                if company not in company_list:
                    company_list.append(company)

        os.system('cls') #show them this
        header = 'PRODUCT: ' + product_list[int(choice)-1]
        print(header.upper())
        print('Number of Companies Involved: '.rjust(30) +
str(len(company_list)))
        print('Number of matching records: '.rjust(30) +
str(len(record_list)))

        print()
        print('ISSUES'.center(50))
        print('------'.center(50))
        issue_list.sort()
        company_list.sort()
        for i in issue_list:
            print(i)
        input('\nPress enter to continue')

setup()

while True:
    os.system('cls') #show them this

    print ('''
----- MAIN MENU -----

Please select from the following options:

1.  Process Complaint Files
2.  Remove Duplicate Complaint Records
3.  Report by Product
```

```
4.  Exit
''')

    user_menu_choice = input('Option#: ')

    if user_menu_choice == '1':
        unpacking()
        continue
    elif user_menu_choice == '2':
        cleanup()
        continue
    elif user_menu_choice == '3':
        report()
        continue
    elif user_menu_choice == '4':
        break
    else:
        print('\nThat is not a valid option.  Please try again.')
        continue
```