Point Of Sale System & Product Info Viewer

Team Members :

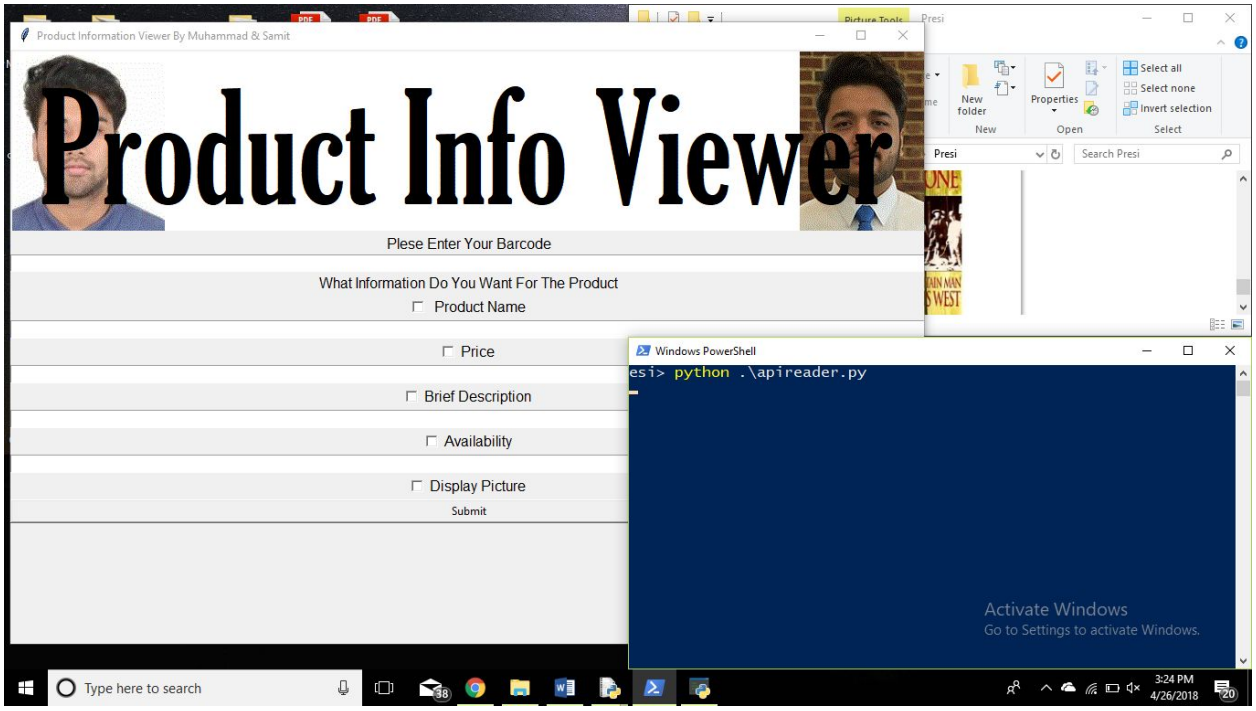
1. Muhammad Ahmed Cheema
2. Samit Basnet

Product Info Viewer:

apireader.py contains a main function and a class window. the class first takes in frame from tkinter which acts as a container for all other widgets. it then calls the grid and create functions. the create function reads the image from the gif file first and then sets it on the grid. it then prints using the label widget and takes in using the entry widget. checkbuttons are then set to a boolean variable and set onto the screen. the enter button takes in the command from the update fill and everytime it is pressed it does all the functions in the update fill. the updatefill function contains all sub functions for the check buttons and makes sure that every time the submit button is pressed it refreshes the data according to the new or present barcode. it first deletes all the data and then re calls the functions to print it on the screen. the functions for the check boxes use the pool manager to define http and then retrieves data from the website in form of json strings. the json strings are then converted and they consist of dictionaries. they contain if and elif statement to delete the information if the checkbox is unchecked and put the information if it is checked using insert. this works the same with all four of the availability, price, description and name functions. the updatepic function saves the image from the url of the image

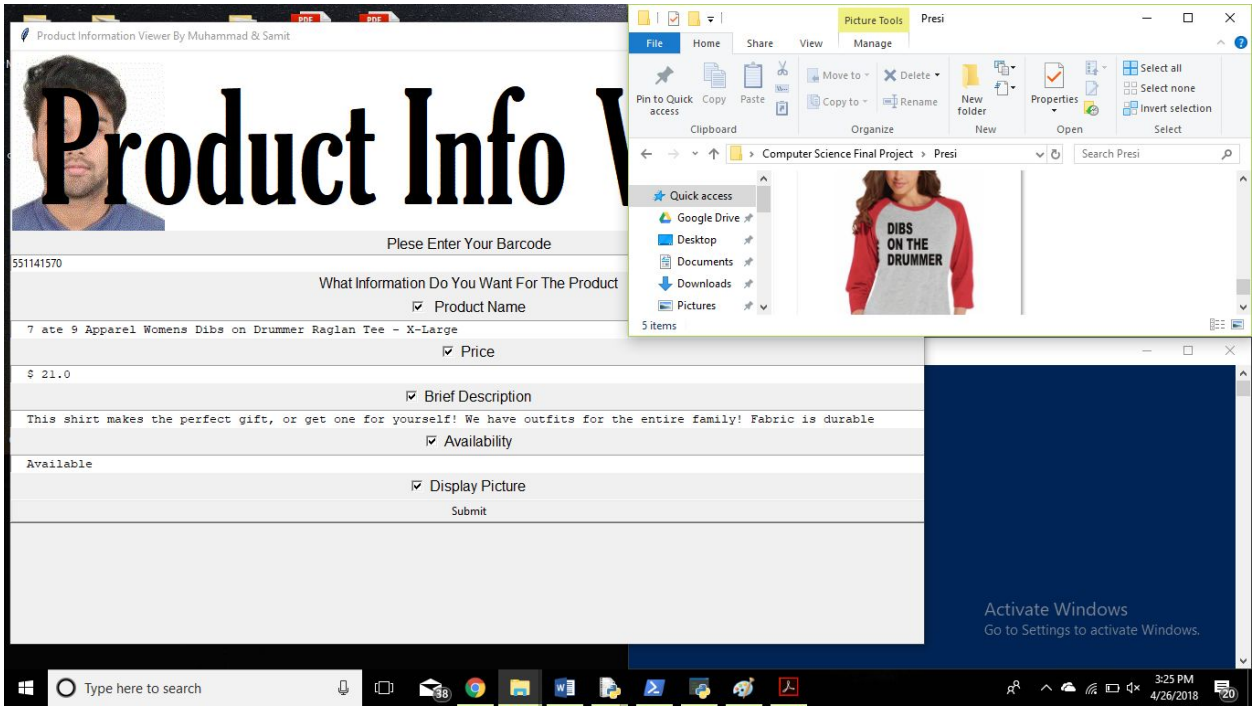
retrieved from the api records. the main function makes the main window frame from tkinter and sets the dimensions to the required pixels using geometry. it also changes the title to the requried one. the code runs as : python3 apireader.py

How To Run The Program:



the program can be started by running it in python by typing python apireader.py. one thing to note is we need the title picture file to be present in the same folder as the program which is a gif file named lol. it then requires the user to choose what things they want to view about that specific product and after they are done checking the checkboxes they should input the

code of the product they want to view and it can be done using a barcode reader or the keyboard (i have attached a list of sample codes that are present in the walmart api). lastly they need to hit enter and they will see the output. the image of that product will be saved with the name “pici” in the same folder that contains the file. after one product and selection is done they just need to enter the new barcode and hit submit. the picture will be updated in the same file and folder.



Problems Encountered & Future Improvements Or Modification:

during the course of this program we did various kinds of testing like in the beginning we only made the program to show the information from the check boxes but it didnt delete when a new code was entered or the checkbox was unchecked so it added the new information to the previous one. we then fixed the program to update the information by deleting the previous one

every time the submit button was pressed with a new code. it was also really hard to put the text boxes and entry widgets in front of the label and check boxes as tkinter had a weird column spacing and everytime we put a column number of 1 the widget went way far and we were also unable to use the pack function. we came across the sticky idea of east and west after some time and before that everything was left aligned. we also tried to add the saved product image and display it on the screen but apparently the saved image was jpg format even though we saved it with the name .gif and so it gave the error that python can't open jpg file. another problem was converting the json strings to usable format and we didn't know about json at first were unable to read a dictionary as it was just a block of strings. moreover there were various other problems that i can mention. however, we are looking forward to make plenty of improvements to our program like giving an option to choose which stores api does the user wants to view and use checkboxes to use that one. we are also looking forward to find a way to change the jpeg to gif and open it or somehow open the jpeg product picture in the main window frame. we are also looking forward to integrate the second part of our project to the first one and instead of hardcoding the data we are looking to use the prices and availability from the api. i think that this program has the potential to be improved forever and ever and we can release updates on timely bases.

Objectives:

1. Scan the barcode of the product to view:
    a. Name of the product
    b. Price of the product
    c. Description of the product
    d. Availability of the product
2. Also to save the save the picture of that object.

External Sources:

we referred to various sources mainly to learn the new modules, tkinter and urllib. we learned how to integrate the button, labels and all other widgets to their commands.

1. https://www.tutorialspoint.com/python3/python_gui_programming.htm
2. https://urllib3.readthedocs.io/en/latest/

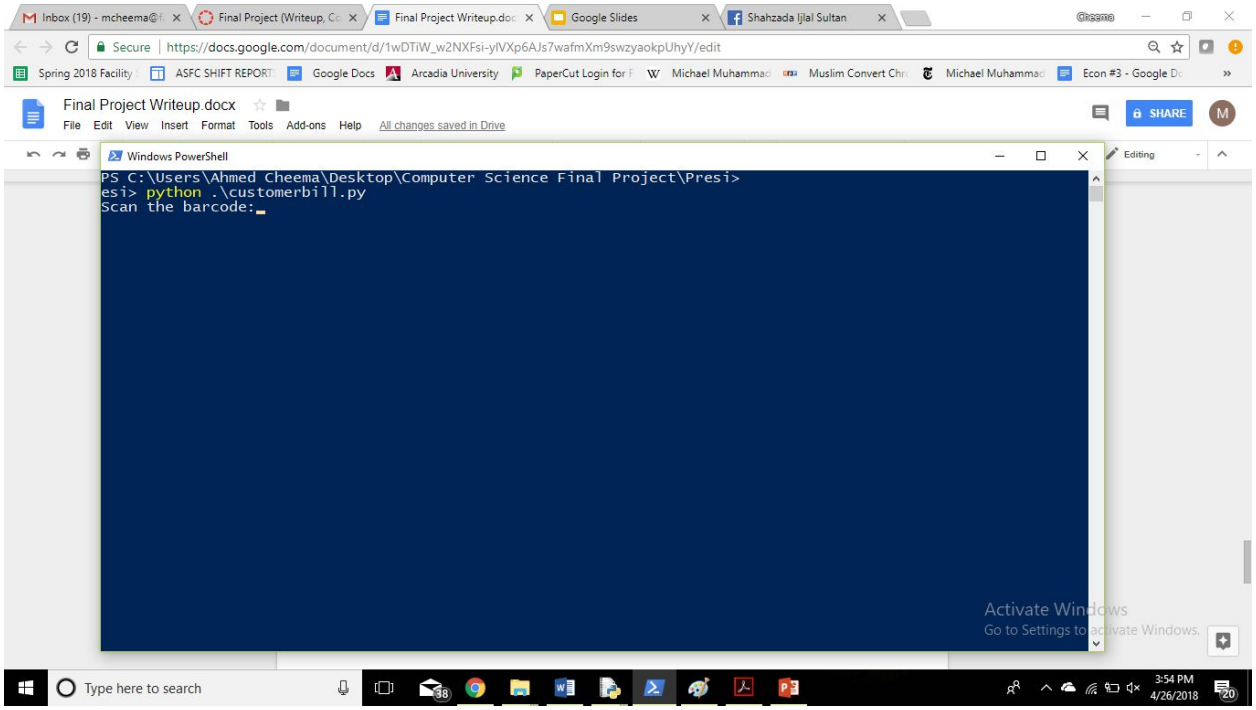
Point Of Sale System:

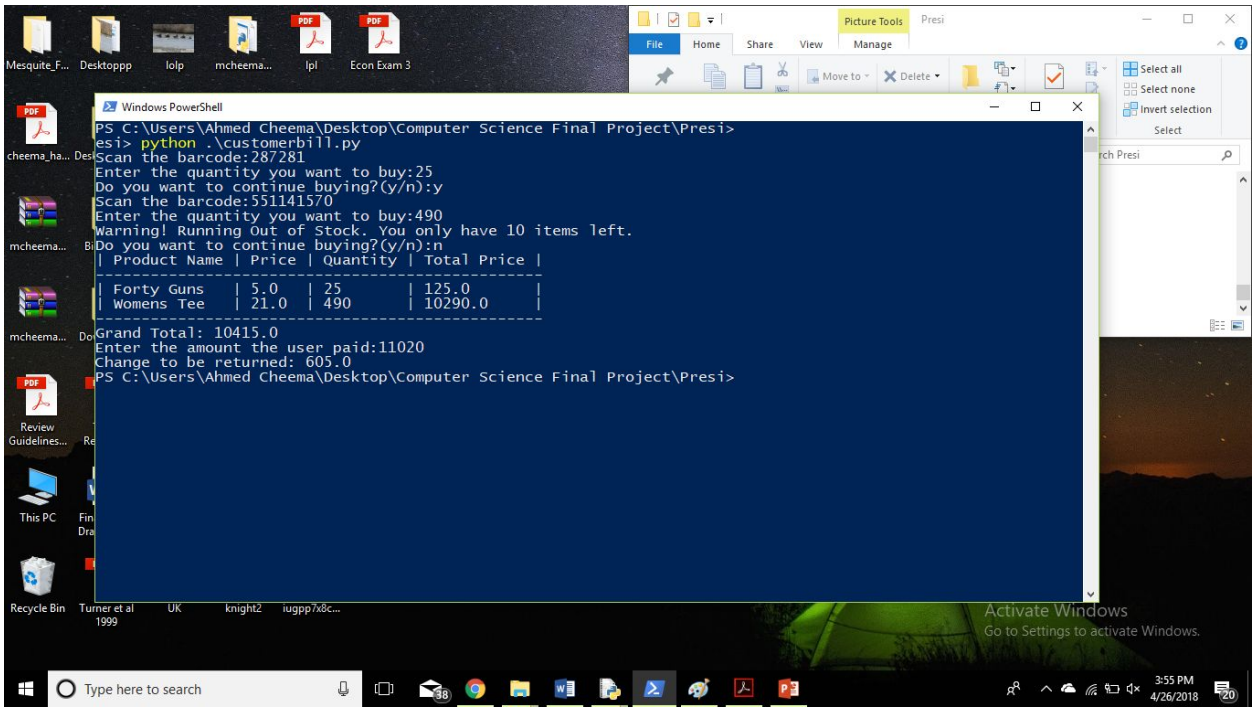
customerbill.py consists of a main function. It takes barcode of a product as an input using the barcode scanner and asks the customer number of quantity of the product they want to buy. The program consists of lists of barcodes and another lists that has names, price and quantity in the store corresponding to that barcode. As the user scans an item, the program adds it to the empty values list of a dictionary inside a separate list called bill.

The program appends name, price and the quantity of the scanned item in the list bill and asks if the customer wants to buy more. When the customer is done with the purchase, it prints out the bill in a table form with the grand total. It asks for the amount paid by the user as an input and shows the amount of change to be returned to the user. This program also keeps track of the quantity of item in the shop. It decreases the quantity purchased by the customer from the quantity in the shop and displays a warning message if the quantity in the shops falls below 10.

How To Run The Program:

1. Run the program using python customerbill.py.
2. Scan the barcode of the product or enter it using the keyboard.
3. Enter the quantity you want to buy.
4. Enter **y** if you want to buy more and **n** if you want the bill.
5. Repeat from scanning the barcode if you chose **y.**
6. If **n**, bill will be shown and enter the amount you paid.
7. It shows the change you will be returned.


Final Project Writeup.docx
Windows PowerShell
PS C:\Users\Ahmed Cheema\Desktop\Computer Science Final Project\Presi>
esi> python .\customerbill.py
Scan the barcode:
Activate Windows
Go to Settings to activate Windows.
Type here to search
3:54 PM
4/26/2018


Windows PowerShell
PS C:\Users\Ahmed Cheema\Desktop\Computer Science Final Project\Presi>
esi> python .\customerbill.py
Scan the barcode:287281
Enter the quantity you want to buy:25
Do you want to continue buying?(y/n):y
Scan the barcode:551141570
Enter the quantity you want to buy:490
Warning! Running Out of Stock. You only have 10 items left.
Do you want to continue buying?(y/n):n
Product Name	Price	Quantity	Total Price
Forty Guns	5.0	25	125.0
Womens Tee	21.0	490	10290.0
Grand Total: 10415.0
Enter the amount the user paid:11020
Change to be returned: 605.0
PS C:\Users\Ahmed Cheema\Desktop\Computer Science Final Project\Presi>
Activate Windows
Go to Settings to activate Windows.
Type here to search
3:55 PM
4/26/2018

Objectives:

1. Scan the barcode of the product
2. Enter the quantity and calculate the total price
3. Generate the grand total and display the bill
4. Take the amount paid by the customer and give the change due
5. Keep track of the quantity of the material left at the store
6. Also notify the shop owner if the inventory level falls below certain level

Modifications For Next version:

1. Integrate this program with the first one so that we do not have to hardcode the information of the product.
2. Print out a hardcopy of the bill for the customer by integrating a printer with python.

Resources And External Tools Used:

None for this program.