



REPORT FOR RETAIL STORE STOCK MAINTENANCE SYSTEM

As a project work for Course
PYTHON PROGRAMMING (INT 213)

Name: Shivani Sharma

Registration Number: 12016247

Program: CSE B.Tech

Semester: Third

School: School of Computer Science and Engineering

Name of the University: Lovely Professional University

Date of submission: 20th NOVEMBER 2021

RETAIL STORE STOCK MAINTENANCE SYSTEM

ABSTRACT:

Retail inventory management is the process of ensuring you carry merchandise that shoppers want, with neither too little nor too much on hand. By managing inventory, retailers meet customer demand without running out of stock or carrying excess supply. Inventory refers to the process of ordering, storing, using, and selling a company's inventory. This includes the management of raw materials, components, and finished products, as well as warehousing and processing of such items. Inventory management is the entire process of managing inventories from raw materials to finished products.

ACKNOWLEDGEMENT:-

I would like to express my special thanks of gratitude to my Prof. Ankita Wadhawan. Who gave me the golden opportunity to do this wonderful project on the topic RETAIL STORE STOCK MAINTENANCE SYSTEM. This also helped me in doing a lot of research and I come to know about so many new things I really thankful to her.

Secondly I would also like to thank my friends and seniors who spent countless hours to listen and provide feedbacks. It helped me increase my knowledge and skills.

INTRODUCTION:-

1.1 Context

This project has been done as part of my course for the CSE at Lovely Professional University. Supervised by Ankita Wadhawan, I have one month to fulfil the requirements in order to succeed the module.

1.2 Motivations

Being extremely interested in everything having a relation with the GUI, the project was a great occasion to give me the time to learn and confirm my interest for this field. That's why I decided to conduct my project around the GUI.

1.3 Idea:-

As a first experience, I wanted to make my project as much didactic as possible by approaching every different steps of the using

GUI and trying to understand it deeply. I choose to take RETAIL STORE STOCK MAINTENANCE SYSTEM as approach. The goal was to know more about inventory and stock maintenance system.

LIBRARIES AND MODULES:

1. Tkinter:

Tkinter commonly comes bundled with Python, using Tk and is Python's standard GUI framework. It is famous for its simplicity and graphical user interface.

2. Tkinter.ttk:

The tkinter.ttk module provides access to the Tk themed widget set, the basic idea for tkinter. ttk is to separate, to the extent possible, the code implementing a widget's behaviour from the code implementing its appearance

SCREENSHOTS:

Inventory System

☐ Coca Cola

☐ Bun

☐ Pizza

☐ white sauce pasta

☐ Veg Biryani

Total

Print

Add

item	price	qty	total
------	-------	-----	-------

RECIPT:

 Inventory System

—

□

×

-----RECIPT-----

Item	PRICE	QTY	TOTAL
Veg Biryani	40.0	2.0	80.0
Total	80.0		

Steps:

First import the libraries.

```
from tkinter import *  
from tkinter import ttk
```

This is for generating receipt:

```
from tkinter import *  
from tkinter import ttk  
  
def print():  
    tott = float(totText.get())  
    top = Toplevel()  
    top.geometry("300x300")  
  
    top.config(bg="white")  
    l = Label(top, text='-----RECIEPT-----')  
    l.pack()  
    l.config(bg="white")  
    heading = Label(top, text='\tItem\tPRICE\tQTY\tTOTAL')  
    heading.pack()  
    heading.config(bg="white")  
  
    for child in listBox.get_children():  
        item = (listBox.item(child, 'values')[0])  
        price = float(listBox.item(child, 'values')[1])  
        qty = float(listBox.item(child, 'values')[2])  
        tot = float(listBox.item(child, 'values')[3])  
        iteml = Label(top, text=f'{item}\t{price}\t{qty}\t{tot}')  
        iteml.config(bg="white")  
        iteml.pack()  
  
    tot = Label(top, text=f'Total\t{tott}')  
    tot.config(bg="white")  
    tot.pack()
```

From this things are added in the list:

```
def show():
    tot = 0
    if (var1.get()):
        price = int(e1.get())
        qty = int(e6.get())
        tot = int(price * qty)
        tempList = [['Coca Cola', e1.get(), e6.get(), tot]]
        for i, (item, price, qty, tot) in enumerate(tempList, start=1):
            listBox.insert("", "end", values=(item, price, qty, tot))

    if (var2.get()):
        price = int(e2.get())
        qty = int(e7.get())
        tot = int(price * qty)
        tempList = [['Bun', e2.get(), e7.get(), tot]]

        for i, (item, price, qty, tot) in enumerate(tempList, start=1):
            listBox.insert("", "end", values=(item, price, qty, tot))

    if (var3.get()):
        price = int(e3.get())
        qty = int(e8.get())
        tot = int(price * qty)
        tempList = [['Pizza', e3.get(), e8.get(), tot]]

        for i, (item, price, qty, tot) in enumerate(tempList, start=1):
            listBox.insert("", "end", values=(item, price, qty, tot))

    if (var4.get()):
        price = int(e4.get())
        qty = int(e9.get())
        tot = int(price * qty)
        tempList = [['white Sauce Pasta', e4.get(), e9.get(), tot]]

        for i, (item, price, qty, tot) in enumerate(tempList, start=1):
            listBox.insert("", "end", values=(item, price, qty, tot))

    if (var5.get()):
        price = int(e5.get())
        qty = int(e10.get())
        tot = int(price * qty)
        tempList = [['Veg Biryani', e5.get(), e10.get(), tot]]

        for i, (item, price, qty, tot) in enumerate(tempList, start=1):
            listBox.insert("", "end", values=(item, price, qty, tot))

    sum1 = 0.0
    for child in listBox.get_children():
        sum1 += float(listBox.item(child, 'values')[3])
    totText.set(sum1)
```

This is the main frame:

```
root = Tk()
root.title("Inventory System")
root.geometry("1000x600")
root.config(bg="sky blue")

totText = StringVar()
balText = IntVar()

Label(root, text="Inventory System", font="arial 22 bold",bg="sky blue").place(x=5, y=10)

var1 = IntVar()
Checkbutton(root, text="Coca Cola", variable=var1,bg="light grey").place(x=10, y=50)

var2 = IntVar()
Checkbutton(root, text="Bun", variable=var2,bg="light grey").place(x=10, y=80)

var3 = IntVar()
Checkbutton(root, text="Pizza", variable=var3,bg="light grey").place(x=10, y=110)

var4 = IntVar()
Checkbutton(root, text="white sauce pasta", variable=var4,bg="light grey").place(x=10, y=140)

var5 = IntVar()
Checkbutton(root, text=" Veg Biryani ", variable=var5,bg="light grey").place(x=10, y=170)
Label(root, text="Total",font="arial 18 bold").place(x=600, y=10)


e1 = Entry(root)
e1.place(x=140, y=50)

e2 = Entry(root)
e2.place(x=140, y=80)

e3 = Entry(root)
e3.place(x=140, y=110)

e4 = Entry(root)
e4.place(x=140, y=140)

e5 = Entry(root)
e5.place(x=140, y=170)

e6 = Entry(root)
e6.place(x=300, y=50)

e7 = Entry(root)
e7.place(x=300, y=80)

e8 = Entry(root)
e8.place(x=300, y=110)

e9 = Entry(root)
e9.place(x=300, y=140)

e10 = Entry(root)
e10.place(x=300, y=170)

tot = Label(root, text="", font="arial 18 bold", textvariable=totText, bg="blue")#all about total
tot.place(x=650, y=60)
```

```
Button(root, text="Add",font="arial 18 bold", command=show, height=2, width=8,bg="red").place(x=10, y=220)

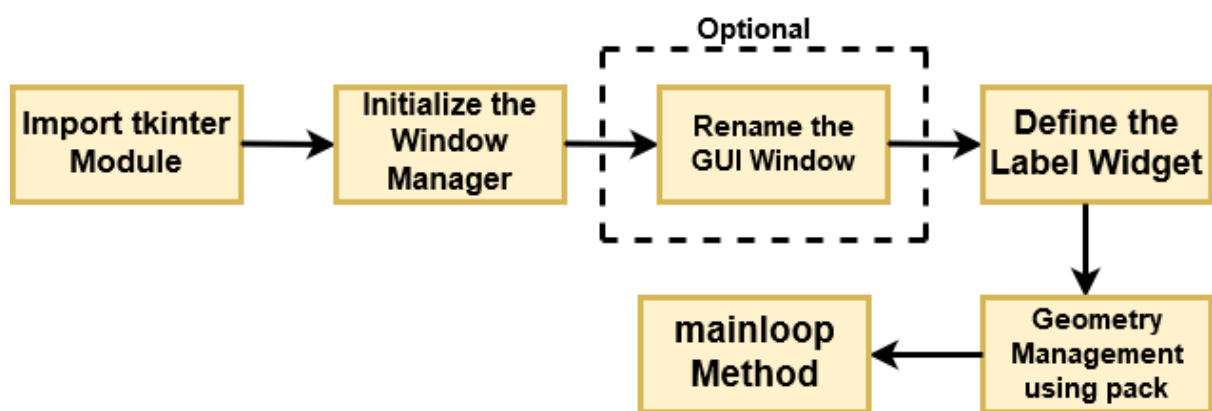
Button(root, text="Print",font="arial 18 bold", command=print, height=2, width=8,bg="yellow").place(x=850, y=120)
cols = ('item', 'price', 'qty', 'total')
listBox = ttk.Treeview(root, columns=cols, show='headings')#all about listbox

for col in cols:
    listBox.heading(col, text=col)
    listBox.grid(row=1, column=0, columnspan=2)
    listBox.place(x=10, y=300)

root.mainloop()
```

Flow Diagram:

Now, let's build a very simple GUI with the help of Tkinter and understand it with the help of a flow diagram.



ALL ABOUT FLOW DIAGRAM:

First, import the key component, i.e., the Tkinter module.

As a next step, initialize the window manager with the `tkinter.Tk()` method and assign it to a variable. This method creates a blank window with close, maximize, and minimize buttons on the top as a usual GUI should have.

Then as an optional step, rename the title of the window as you like with `window.Title(title_of_the_window)`.

Next, make use of a widget called Label, which is used to insert some text into the window.

Then, make use of Tkinter's geometry management attribute called `pack()` to display the widget in size it requires.

Finally, as the last step, use the `mainloop ()` method to display the window until you manually close it. It runs an infinite loop in the backend.

Conclusion:-

It is my hope that this document will be of huge help with understanding of our little project as I have used a different approach which has proved beneficial for me and easy for me to understand the vast ocean that is GUI in python. I have reached the maximum accuracy of 95% after data cleaning but I will work forward to increase this accuracy little by little.

REFERENCES:-

To conduct this project the following tools have been used:

- Jupyter notebook and IDLE

1.1 Geeks for Geeks

We have used this side for our basic knowledge gain of the methods that will be used in the project

<https://www.geeksforgeeks.org/python-gui-tkinter/>

1.2

We have used this site for solving our different problems which has occurred during this project.

<https://stackoverflow.com/questions/tagged/python>