

Project : Color Shop

Group name : Pythoners

Team members:

Hamza Ishaq

Ali And Arqum

Github links:

[Hamza](#)

[Muhammad-Arqum](#)

## Project Description :

This app implements a Point of Sale (POS) system using the customtkinter and tkinter libraries for the graphical user interface, openpyxl for Excel file operations, and PIL for image handling. It features a login system, product management, cart functionality, and printing bill capabilities.

**Important Modules:**

```
pip install customtkinter  
pip install CTkmessagebox  
pip install CTkListbox  
pip install openpyxl
```

## Imports :

```
1  from tkinter import *  
2  from tkinter import ttk  
3  
4  import customtkinter  
5  from CTkListbox import *  
6  from CTkMessagebox import CTkMessagebox  
7  from customtkinter import *  
8  from PIL import Image  
9  import tkinter  
10 import openpyxl  
11 from openpyxl import Workbook  
12 import datetime  
13
```

Other Modules are pre installed in python

## **Methods Used In Program :**

### **Add\_to\_cart**

**Adds the specified product and quantity to the shopping cart.**

**Validates product availability against the Excel sheet data/products.xlsx, updates the product stock, and displays**

**an error message if the  
product is out of stock or not  
found.**

```

68 def add_to_cart():
69
70     #first we will we will take product and quantity from th
e entry boxes with .get
71     product = enter_product.get().strip()
72     # where I will use try and except method to produce an
error if you user enters a string
73
74     quantity = int (enter_quantity.get())
75
76     #now I will loop through the rows of the sheet and add
the data to the dictionary
77
78     for row in sheet.iter_rows():
79
80
81         #now I will get the product name price and quantity f
rom the excel file
82
83         product_name =row[0].value
84         product_price =row[1].value
85         product_stock =row[2].value
86
87
88         #here i will use if conditions to check the product
89
90         if product == product_name:
91
92             # now will check if product stock is available
93
94             if product_stock >= quantity:
95
96                 #using another if so that it only works when stoc
k is available
97                 #now i will total the price of the product_stock
98
99                 total_price = product_price * quantity
100
101                 #now I will format the product detail as a string
102                 product_details=f"{product_name
}
x
{quantity}
x
{product_price}
=
{total_price}"
103                 #now I will add this formatted string into my car
t list box
104                 cart_listbox.insert("end",product_details)
105                 #now I will update the stock in my excel file
106                 sheet.cell(row, 2,product_stock - quantity)
107                 #now I will save the changes in my excel file and
break the loop
108                 workbook.save(path)
109
110                 cart_listbox.insert("end", product_details)
111                 #using .delete to delete text from entry boxes
112                 enter_product.delete(0, "end")
113                 enter_quantity.delete(0, "end")
114                 break
115                 #give same code by using now I will create an else st
atement that will show an error if product stock is not e
nough
116
117                 #using CTkMessageBox module i imported to show an error
118
119                 else:
120                     CTkMessageBox(title="Error", message=
"Product is out of stock", icon="warning")
121                     enter_product.delete(0, "end")
122                     enter_quantity.delete(0, "end")
123                     break
124
125                     elif product == "" and quantity == "":
126                         CTkMessageBox(title="Error", message=
"Please enter Product Name and password", icon="warning")
127
128
129                     elif product == "":
130                         CTkMessageBox(title="Error", message=
"Please enter Product Name", icon="warning")
131
132                     elif quantity == "":
133                         CTkMessageBox(title="Error", message=
"Please enter Quantity", icon="warning")
134
135                     else:
136                         #Error when Product is not found
137                         CTkMessageBox(title="Error", message=
"Product Not Found", icon="warning")
138                         enter_product.delete(0, "end")
139                         enter_quantity.delete(0, "end")
140
141

```

**Calculate\_total:**

**Calculates the total amount for the items added to the cart. It iterates over each item in the cart, extracts the price and quantity, calculates the total cost, and updates the total amount display.**



```
169
170 def calculate_total():
171     #first make a total amount var and gave it 0
172     total_amount = 0
173     #now i will loop through the cart
174     for item in cart_listbox.get(0,"end"):
175
176         #where I will create a variable called item parts and
        I will use items dot split to convert product detail str
        ain into different strings its like it will create an lis
        t with different strings
177
178         product, price, quantity = item.split(" ")
179         # Convert the price and quantity to float and int
        respectively
180         price = float(price)
181         quantity = int(quantity)
182
183         # Multiply the price and quantity and add it to the total
        total_amount += price * quantity
184         # Display the total in a label
185         total_label.config(text=f"Total: {total}")
186
187
```

## **Login:**

**Handles the login functionality.**

**It verifies entered username and password against predefined credentials. If the credentials match, it opens a new 'Admin Page' window.**

**Provides error messages for empty input fields or incorrect credentials. New admin window has a treeview connected to an excel file.**

```

192 def login():
193     #first I will get the username and password from the en
    try boxes
194
195     username = enter_username.get()
196     password = enter_password.get()
197
198     #now I will check if the username and password are corr
    ect
199     if username == "admin" and password == "admin":
200         #if correct creating a new top level window in root
        window with argument root
201         admin = customtkinter.CTkToplevel(root)
202
203         admin.geometry("800x600")
204         admin.maxsize(800,600)
205         admin.title("Admin Page")
206         #this WM transient command will put top level window
        in front of my main window
207         #thank to stackoverflow i found this command
208         admin.wm_transient(root)
209
210
211 #now I WILL configured the rows and columns in my admin w
    indow
212     admin.columnconfigure(0,weight =2,uniform="a")
213     admin.columnconfigure(1,weight =3,uniform="a")
214     admin.rowconfigure(0,weight =1,uniform="a")
215     admin.rowconfigure(1,weight =3,uniform="a")
216
217
218
219     #first i will create the admin heading frame in admin w
    indow
220
221     admin_heading_frame = customtkinter.CTkFrame(admin,
        width=100, height=100)
222     #now i will place the heading frame with grid
223     admin_heading_frame.grid(row=0, column=0, columnspan=
        3,padx=3, sticky=tkinter.EW)
224
225     #now i will give rows and columns to the admin head
    ing frame
226
227     admin_heading_frame.columnconfigure(1,weight =1,
        uniform="a")
228     admin_heading_frame.rowconfigure(1,weight =1,uniform=
        "a")
229
230     admin_heading_Label= customtkinter.CTkLabel(
        admin_heading_frame ,text='HELLO ADMIN',
231         font=('times new roman',45,'bold'),
232         text_color='yellow',
233         image=img,
234         # i imported this img at the start of the code
        compound=LEFT)
235
236
237     admin_heading_Label.grid(row=0, column=0, columnspan=
        2,padx=5, sticky=tkinter.NSEW)
238
239
240
241
242
243
244     product_frame = customtkinter.CTkFrame(admin, width=
        200, height=100)
245
246     # placing my frames in admin window with .grid
247
248     product_frame.grid(row=1, column=0, columnspan=1,padx
        =5, sticky=tkinter.NSEW)
249
250     product_frame.columnconfigure(0,weight =1,uniform="a"
        )
251     product_frame.rowconfigure(0,weight =1,uniform="a")
252     product_frame.rowconfigure(1,weight =1,uniform="a")
253     product_frame.rowconfigure(2,weight =1,uniform="a")
254     product_frame.rowconfigure(3,weight =1,uniform="a")
255
256
257
258
259
260 #now I will create entry boxes and buttons
261 #buttons
262

```

```

260 #now I will create entry boxes and buttons
261     #buttons
262
263
264
265 #entry boxes
266
267     enter_product_name = customtkinter.CTkEntry(
product_frame,
268         placeholder_text= "Enter product name",
269         width=500,
270         height=60,
271         font=("helvetica",24),
272         text_color="#F5DD90",
273         placeholder_text_color="yellow",
274         corner_radius=200)
275
276
277     enter_product_quantity = customtkinter.CTkEntry(
product_frame,
278         placeholder_text= "Enter product quantity",
279         width=500,
280         height=60,
281         font=("helvetica",24),
282         text_color="#F5DD90",
283         placeholder_text_color="yellow",
284         corner_radius=200)
285
286
287
288
289     enter_product_price = customtkinter.CTkEntry(
product_frame,
290         placeholder_text= "Enter Product Price",
291         width=500,
292         height=60,
293         font=("helvetica",24),
294         text_color="#F5DD90",
295         placeholder_text_color="yellow",
296         corner_radius=200)
297
298
299     save_product = customtkinter.CTkButton(product_fra
, text ="Save Product",
300         width =120,
301         height =60,
302         font=("helvetica",19),
303         text_color="black",
304         fg_color="yellow",
305         hover_color="#c2b84e",
306         corner_radius=200,
307
308
309
310
311
312     # placing the widgets in the product_add_frame
313     enter_product_name.grid(row=0, column=0, padx=5, pady
=5,)
314     enter_product_price.grid(row=1, column=0, padx=5,
pady=5,)
315     enter_product_quantity.grid(row=2, column=0, padx=5,
pady=5,)
316     save_product.grid(row=3, column=0, padx=5, pady=5,)
317
318
319
320
321
322
323
324
325     exel_file_frame = customtkinter.CTkFrame(admin, width
=100, height=100)
326
327
328
329
330
331
332
333     # this excel file frame will contain my excel file an
d show it in my admin window
334
335     exel_file_frame.grid(row=1, column=1,padx=5, sticky=
tkinter.NSEW)
336
337
338

```



```

341 # setting up the function that will run when the save button is clicked
342
343 # configuring excel file frame
344
345 exel_file_frame.columnconfigure(0, weight=1)
346 exel_file_frame.columnconfigure(1, weight=3)
347 exel_file_frame.columnconfigure(2, weight=1)
348 exel_file_frame.rowconfigure(0, weight=1)
349
350
351
352 # so here I will use TTK frame from TTK module
353 tree_frame = ttk.Frame(exel_file_frame, width=100, height=500)
354
355 # will it a grid
356 tree_frame.grid(row=0, column=0, columnspan=2, padx=5, pady=5, sticky=tkinter.NSEW)
357
358 tree_frame.columnconfigure(0, weight=1)
359 tree_frame.rowconfigure(0, weight=1)
360
361 # now i will create a treeview in tree frame
362
363
364
365 # here I will create a scroll bar to connect to our tree view to scroll
366
367 tree_scroll = ttk.Scrollbar(tree_frame,)
368 # here I will pack it to the right side
369 tree_scroll.grid(row=0, column=1, sticky=tkinter.NSEW)
370 )
371 # now I will create a tuple named colm and i will put names of my columns in excel file
372 colm = ("name", "price", "stock")
373
374
375
376 #now I will create a treeview with the columns i created
377
378 tree_view = ttk.Treeview(tree_frame,
379 show="headings",
380 yscrollcommand=tree_scroll.set,
381 columns=colm, height=13 )

```

7:09

2.15 KB/s 42%

## ← Color\_Shop.py



```

375 #now I will create a treeview with the columns i created
376
377 tree_view = ttk.Treeview(tree_frame,
378 show="headings",
379 yscrollcommand=tree_scroll.set,
380 columns=colm, height=13 )
381
382
383 # now I will configure the size of the columns of my tree view
384
385
386
387 tree_view.column("name", width=50)
388 tree_view.column("price", width=35)
389 tree_view.column("stock", width=25)
390
391
392
393
394
395
396 # now i will pack it
397
398 tree_view.grid(row=0, column=0, padx=5, pady=5, sticky=tkinter.NSEW)
399
400 tree_scroll.config(command=tree_view.yview)
401
402
403
404 # now I will use openpyxl modular imported before
405
406 # now I will make a function to load data
407 def load_data():
408
409
410
411 # now I will create a list with tuples of all the data
412
413 list_values = list(sheet.values)
414 print(list_values)
415 #now I will loop through the list and add the data to the treeview
416 for col_name in list_values[0]:

```

## ← Color\_Shop.py



```
# now I will create a list with tuples of all the data
411
412     list_values = list(sheet.values)
413     print(list_values)
414     #now I will loop through the list and add the data to
the treeview
415     for col_name in list_values[0]:
416         # now I will specify the name of my heading in my tre
e view
417         tree_view.heading(col_name, text=col_name)
418         for value_tuple in list_values[1:]:
419             tree_view.insert('', tkinter.END, values=
value_tuple)
420
421
422     # now I will loop through the list values and add the
data to the treeview
423
424
425
426     load_data()
427
428
429     # let's add two nested functions
430
431     def save_product():
432
433         #first we will get the product name, quantity and p
rice from the entry boxes
434         product_name = enter_product_name.get()
435         product_stock = enter_product_quantity.get()
436         product_price = enter_product_price.get()
437
438         # now I will load the excel file same as I did befor
e in load data function
439
440         # right now I am inserting data into Excel sheet
441         path = "data/products.xlsx"
442         workbook = openpyxl.load_workbook(path)
443         #now I will select the sheet I want to use
444         sheet = workbook.active
445
446         # I will give the row values
447         row_values = ["name", "price", 'stock']
448
449         # now I will append these R0ws into sheets
450
```

## ← Color\_Shop.py



```
448
449     # now I will append these R0ws into sheets
450
451     sheet.append(row_values)
452     # now I will save the workbook
453     workbook.save(path)
454
455     # adding this data into tree view
456
457     tree_view.insert("", tkinter.END, values=row_values
)
458
459
460     #now i will create save button
461
462
463
464
465
466     elif username == "" and password == "":
467         CtkMessageBox(title="Error", message=
"Pleace enter username and password", icon="warning")
468
469
470     elif username == "":
471         CtkMessageBox(title="Error", message=
"Pleace enter username", icon="warning")
472
473     elif password == "":
474         CtkMessageBox(title="Error", message=
"Pleace enter password", icon="warning")
475
476     else:
477         CtkMessageBox(title="Error", message=
"Invalid username or password", icon="warning")
478
479
480 # login function then now I will work on print bill funct
ion
481
```

**Print\_bill:**

**Generates a bill for the customer, including the customer's name, phone number, date-time of purchase, and item details. It calculates the total amount and displays a thank you message at the end.**

```
482
483 def print_bill():
484
485     # i will get the customer name from the customer_name_e
ntry widget
486     customers_name = customer_name.get()
487     # i will get the customer number from the customer_num
ber_entry widget
488     customers_number = customer_number.get()
489     # i will get the current date and time as a string
490     date_time = datetime.datetime.now().strftime(
"%Y-%m-%d %H:%M:%S")
491     # i will clear the bill_textbox widget
492     bill_textbox.delete("1.0", "end")
493     # now i will insert the customer name, number, and da
te and time into the bill_textbox widget
494     bill_textbox.insert("end", f"Customer Name: {
customer_name}\n")
495     bill_textbox.insert("end", f"Customer Number: {
customer_number}\n")
496     bill_textbox.insert("end", f>Date and Time: {date_time}
\n")
497     # now i will insert a separator line into the bill_te
xtbox widget
498     bill_textbox.insert("end", "-" * 40 + "\n")
499     # loop through the items in the cart listbox
500     for item in cart_listbox.get(0, "end"):
501
502         # now i will insert the item into the bill_textbox widget
503         bill_textbox.insert("end", f"{item}\n")
504         # now i will again insert a separator line into the b
ill_textbox widget
505         bill_textbox.insert("end", "-" * 40 + "\n")
506         # now i will get the total amount from the total_labe
l widget
507         total_amount = total_label.cget("text")
508         # now i will insert the total amount into the bill_te
xtbox widget
509         bill_textbox.insert("end", f"{total_amount}\n")
510         # now i will insert a thank you message into the bill
_textbox widget
511         bill_textbox.insert("end",
"Thank you for shopping with us!\n")
```



## Clear\_cart

**Clears all items from the cart listbox and resets the total label to zero. Displays a success message upon completing the operation.**

```
513
514
515 def clear_cart():
516     # clear the cart_listbox widget
517     cart_listbox.delete(0, "end")
518     # clear the total_label widget
519     total_label.config(text="Total: 0")
520
521     # show a success message
522     CtkMessageBox(title="Success", message=
"Cart cleared successfully", icon="check")
523
524
525
```

# Exit\_program

Prompts the user with a confirmation message before exiting the application. If the user confirms, it destroys the root window, effectively closing the application.

```
149
150 def exit_program():
151     # Show some retry/cancel warnings
152     msg = CtkMessagebox(master=root,
153         title="Are You Sure!",
154         message="Do you want exit?",
155         icon="warning",
156         option_1="Yes",
157         option_2= "No")
158     yes_or_no = msg.get()
159     if yes_or_no == "Yes":
160         root.destroy()
161     else:
162         print("press yes to exit")
163
164
```

For Frontend Code check color shop .py file.

Output:

Main Window :

POS System

**KAMAL PAINT PORTAL**

Enter Username

Enter Password

Login

Order Details

Enter product name

Enter Quantity

add to cart

Total

Customer Details

Customer Name

Customer Number

Print Bill

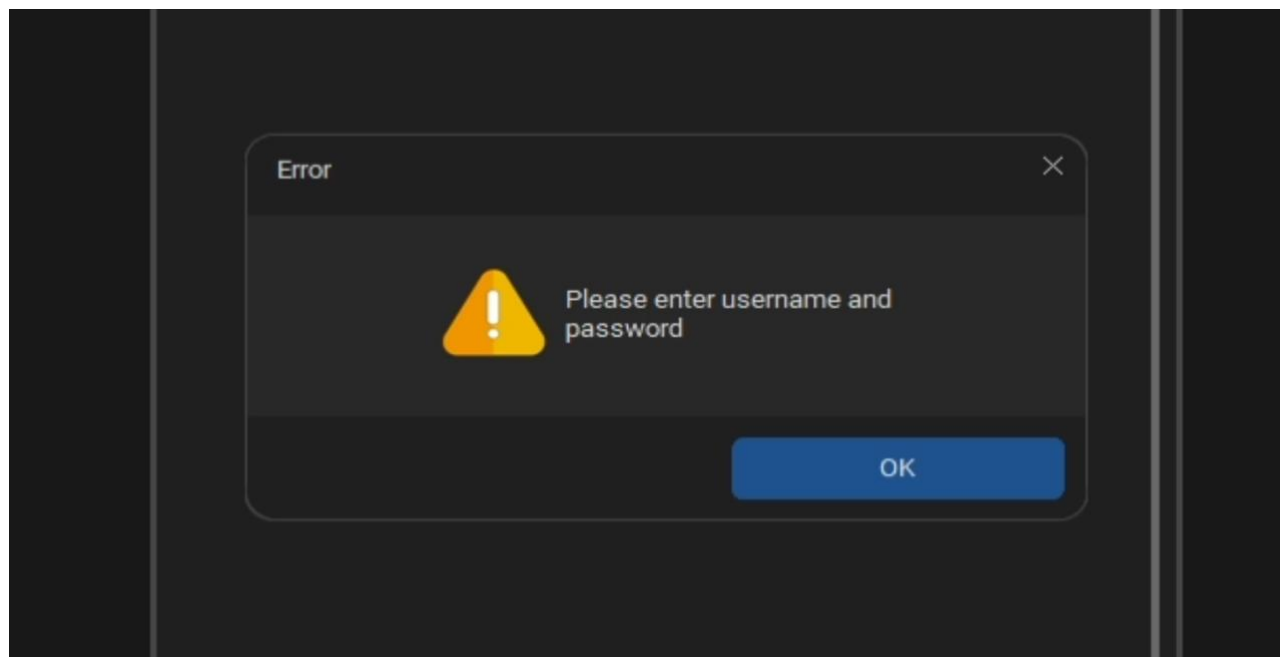
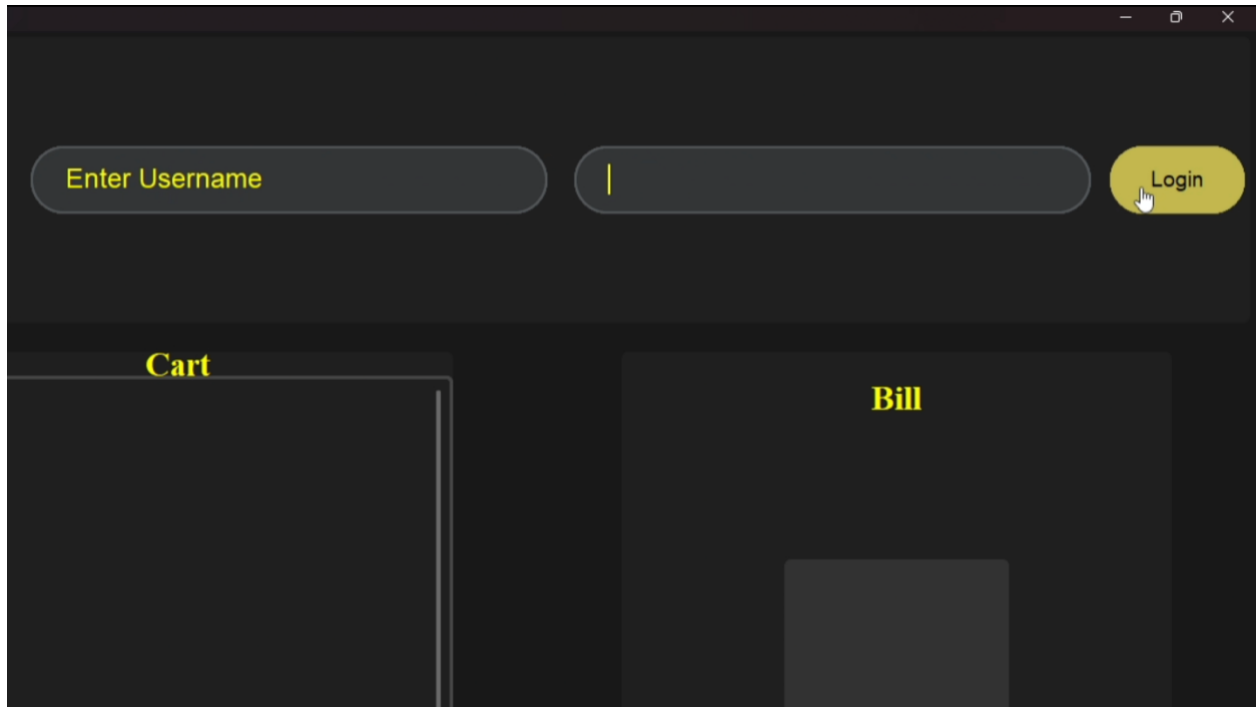
**Cart**

**Bill**

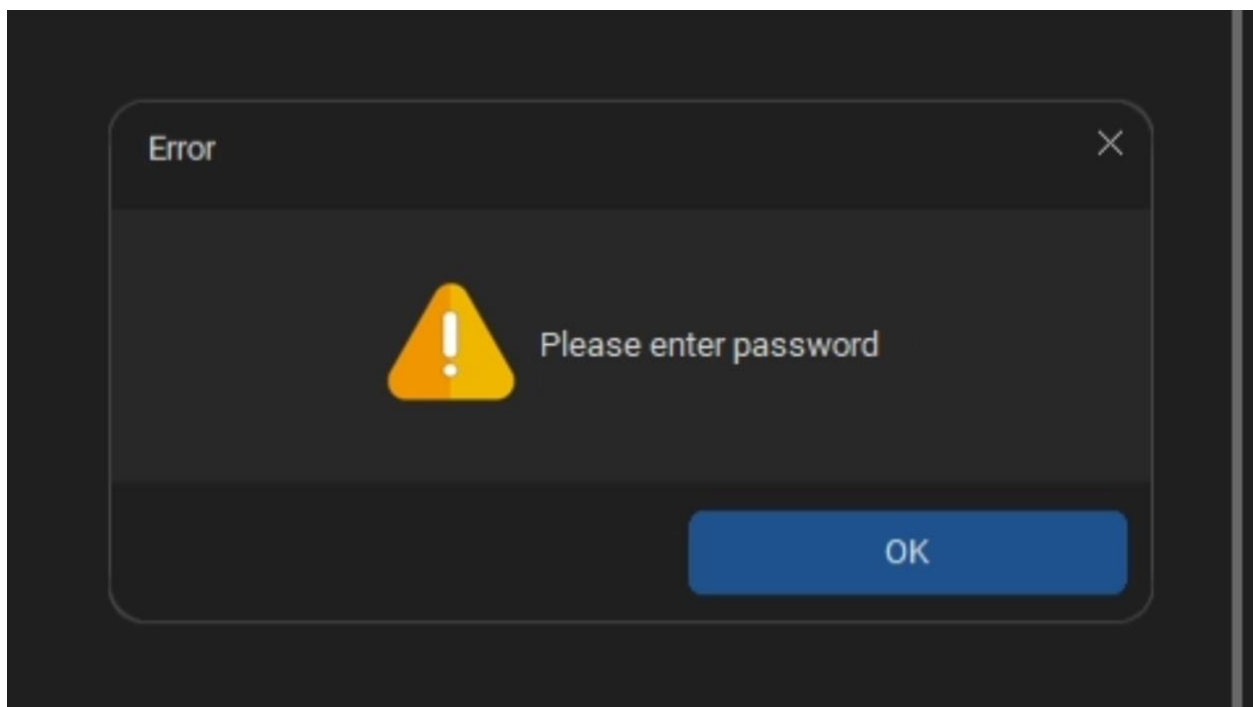
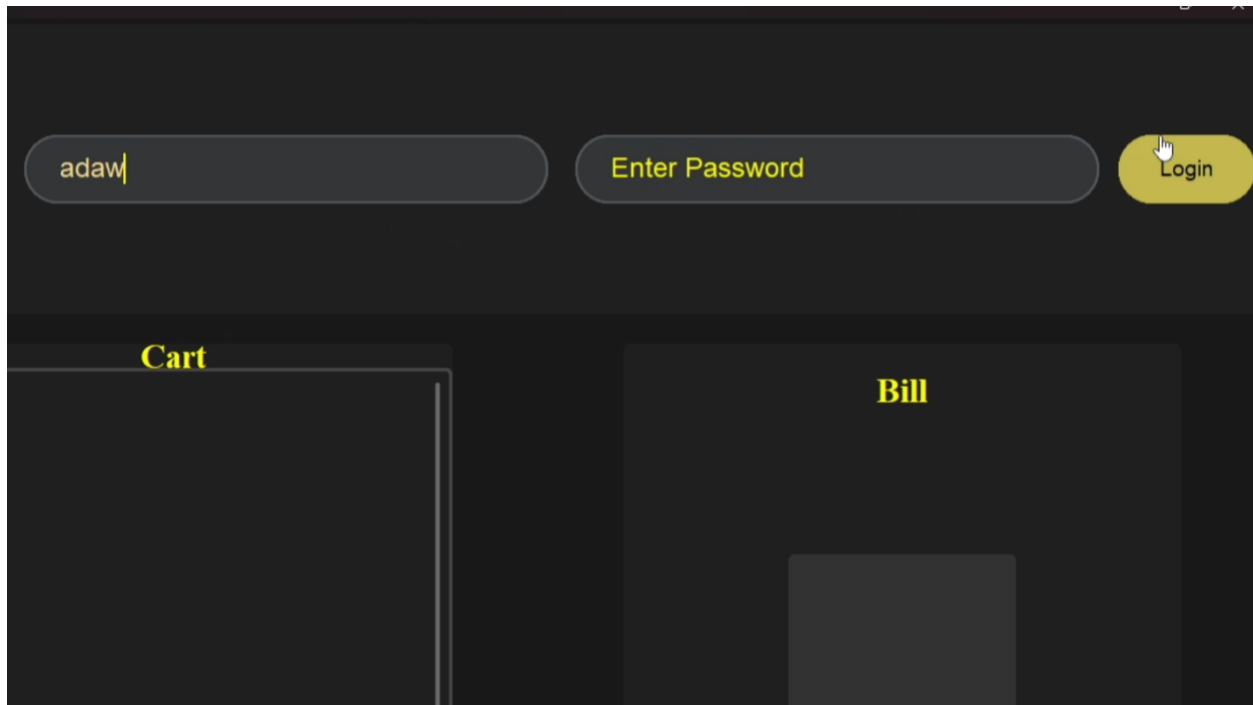
Clear Cart

Exit

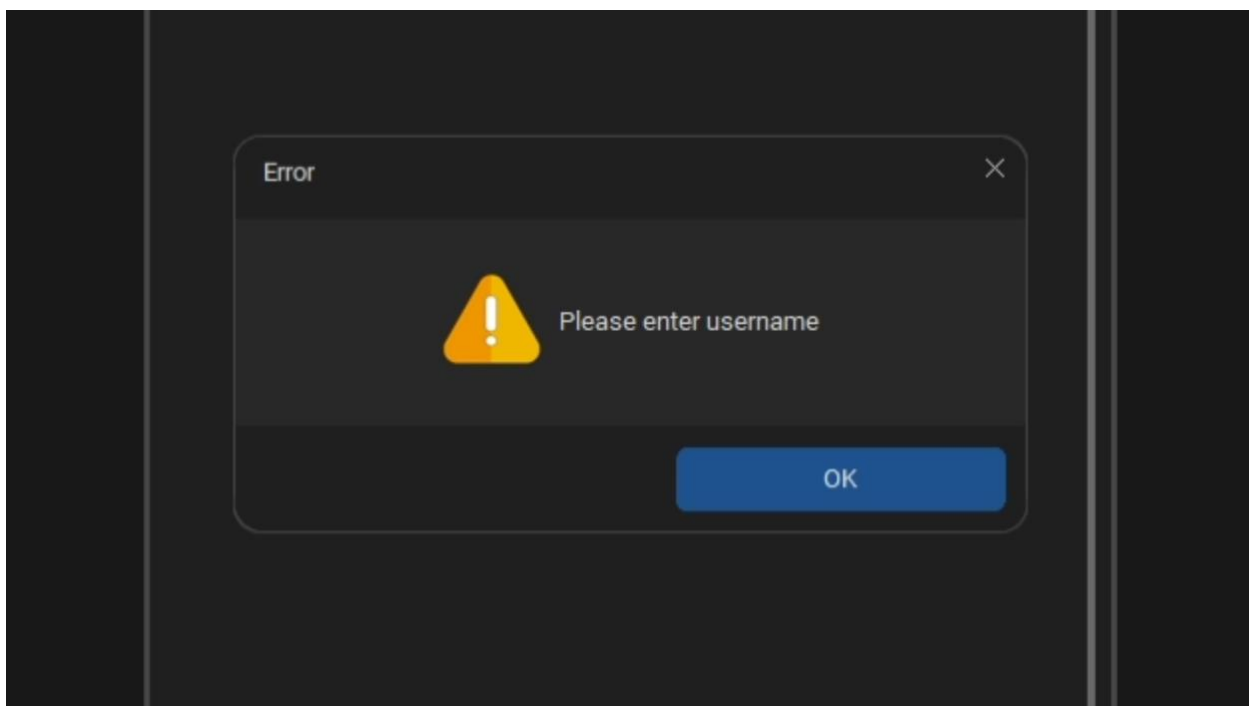
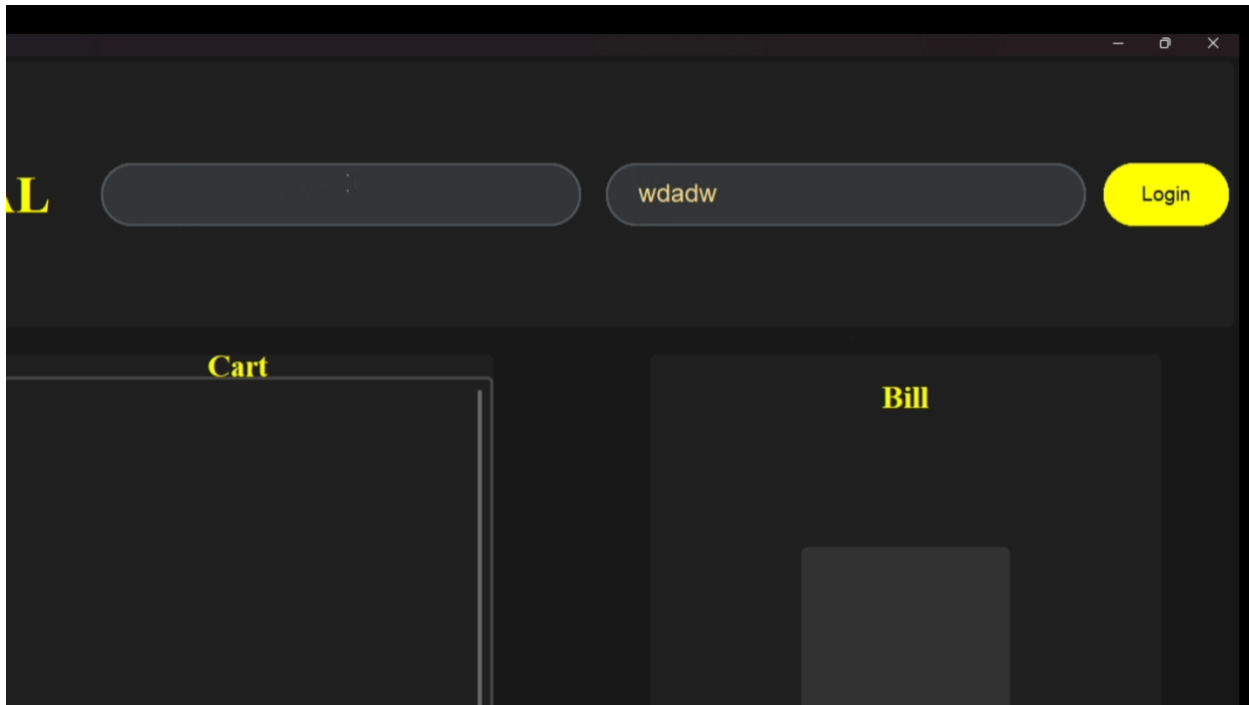
# Error at empty username and password



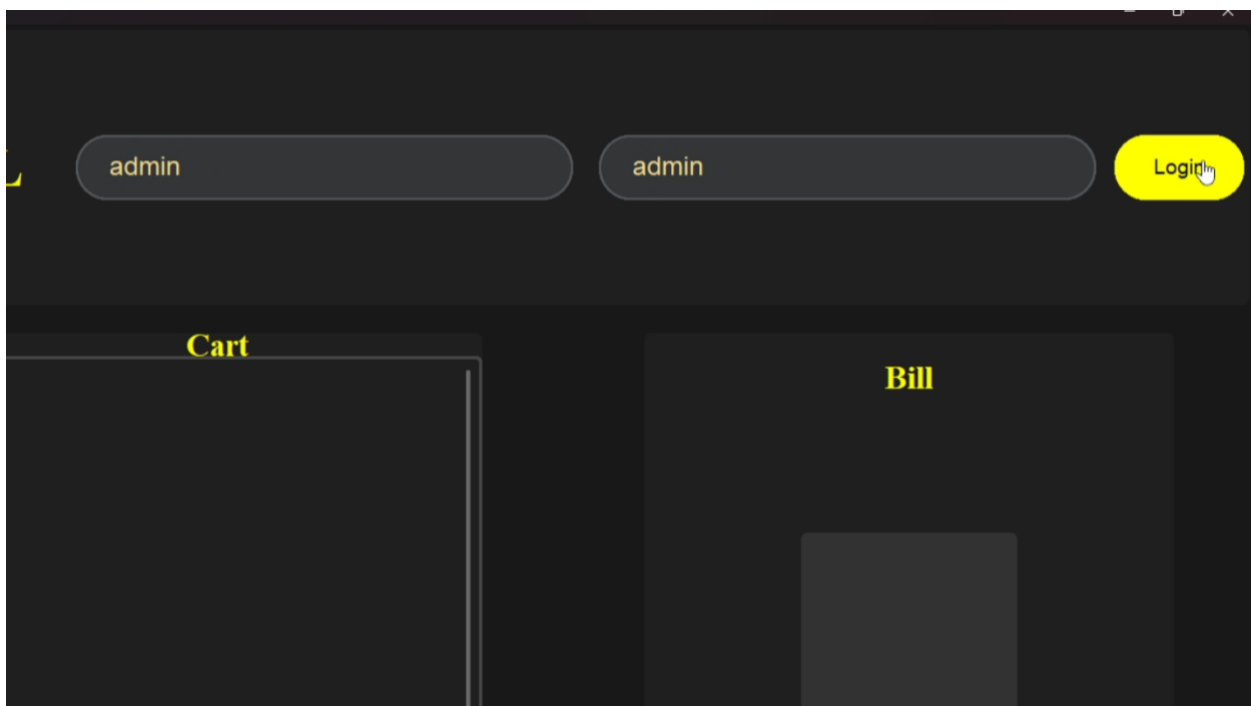
## empty password error

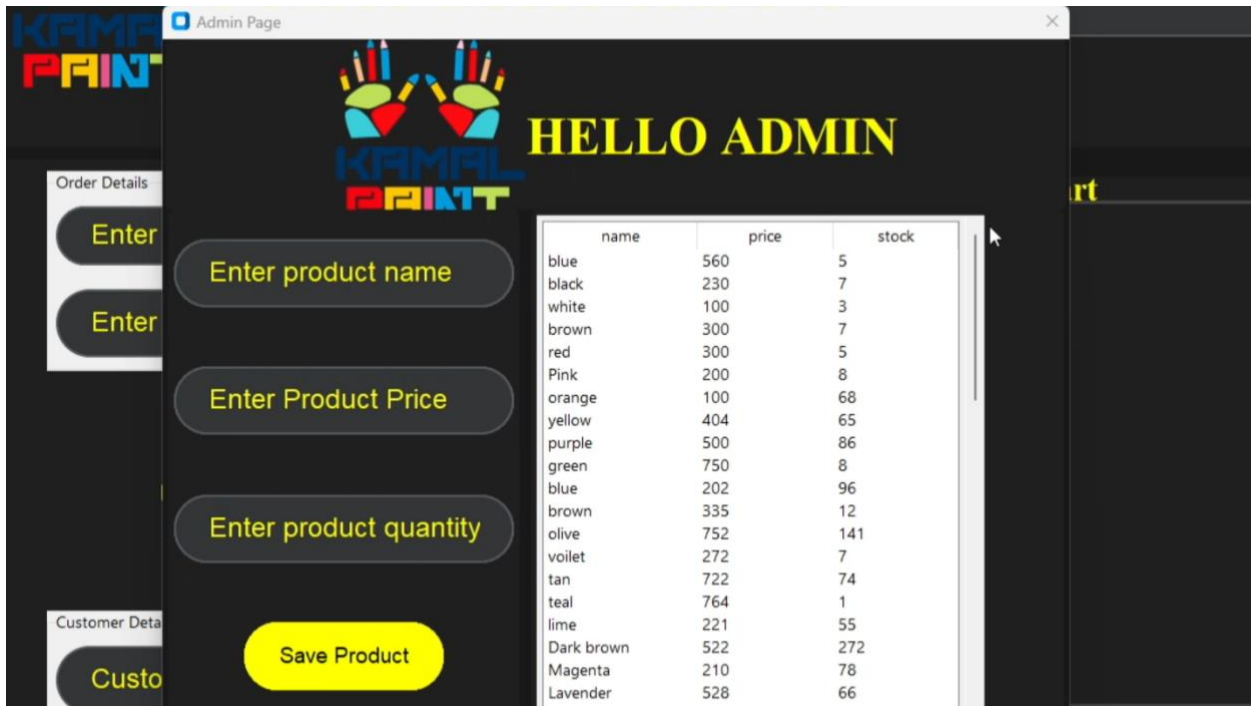


## empty username error

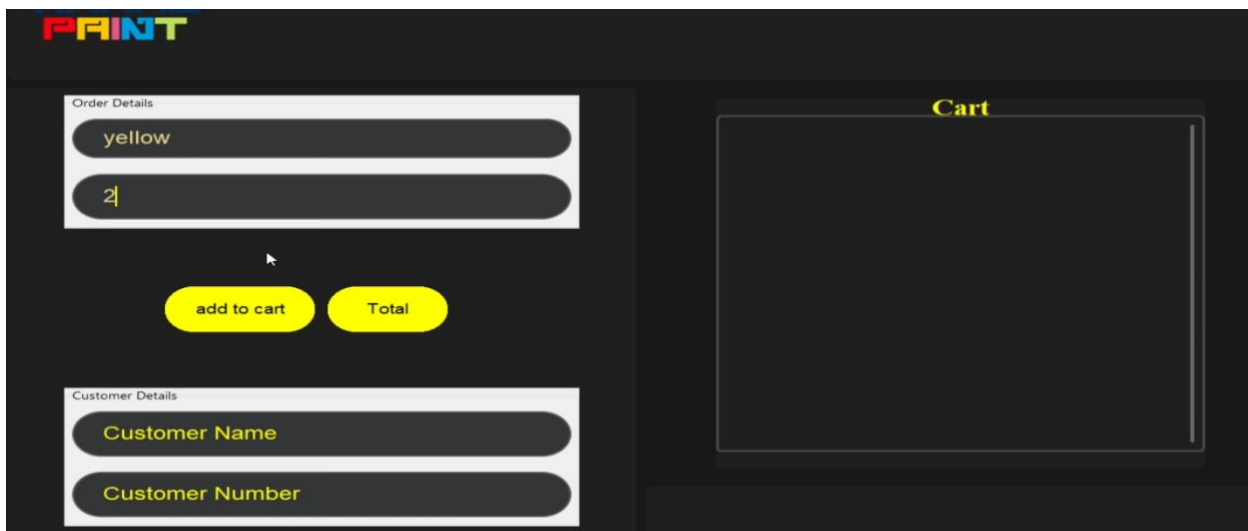


**correct password opens a new  
windows with inventory  
information**





## Adding item into the cart





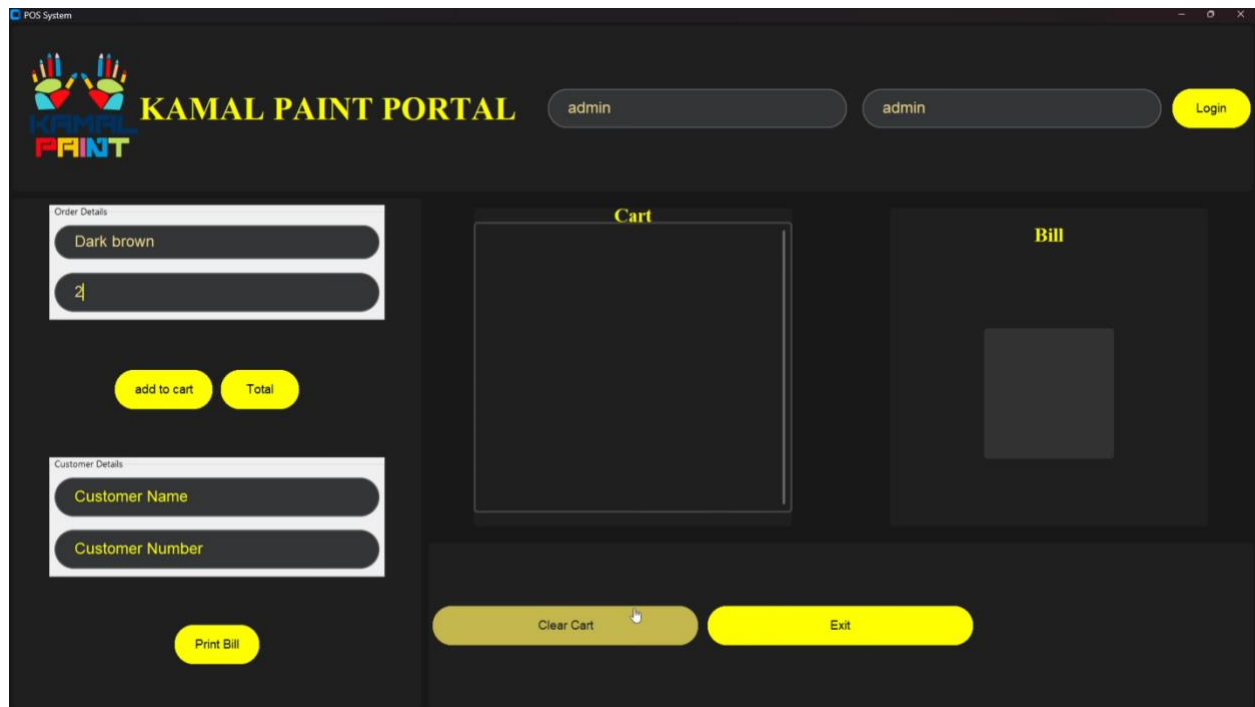
## Cart

yellow × 2 × 404 = 808

## Cart

yellow	x	2	x	404	=	808
white	x	1	x	100	=	100
ark brown	x	2	x	522	=	10

**clearing cart with cart button**



**and an exit button will destroy the window  
and end the program**

**THANKS**

**MORE UPDATES COMING SOON**

