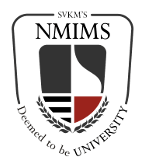
****

**Mini Project**

**Programming Laboratory III** (Python Language)

**Student Choice Stationary Inventory Management**

**Submitted By: Guided By:**

Mohit Bohra (N291) Prof. Dhanajay Joshi

Prasoon Taylor (N275)

Priyanshi Mangal (N293)

Sanya Motwani (N287)

**Code:**

# import all modules

from tkinter import \*

import sqlite3, datetime, math, os, random

from tkinter import messagebox

import threading

conn = sqlite3.connect("G:\mini3\store.db")

c = conn.cursor()

# date

# date = datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S")

date = datetime.datetime.now().date()

# temporary lists .......

products\_list = []

products\_price = []

products\_quantity = []

product\_id = []

# list for labels....

label\_lists = []

username = 'admin'

class Application:

def \_\_init\_\_(self, master):

self.master = master

# creating the frames

self.left = Frame(master, width =700, height =768, bg='white')

self.left.pack(side=LEFT)

self.right = Frame(master, width=666, height=768, bg='lightblue')

self.right.pack(side=RIGHT)

# components

self.heading = Label(self.left, bg='white', text='Students Choice Stationary', font=('arial 40 bold'))

self.heading.place(x=0, y=0)

self.user = Label(self.right, text= 'Admin: ' + str(username), font='arial 16 bold', fg='white', bg='lightblue')

self.user.place(x=200, y=0)

self.logout = Button(self.right, text='Logout', cursor='hand2', width=22, height=2, bg='lightblue', fg='white', command=self.logout)

self.logout.place(x=500, y=0)

self.date\_l = Label(self.right, text="Date: " + str(date), font='arial 16 italic', bg='lightblue', fg='white')

self.date\_l.place(x=0, y=0)

# table invoice ......

self.tproduct = Label(self.right, text= 'Products', font=('arial 18 bold'), bg='lightblue', fg='white')

self.tproduct.place(x=0, y=60)

self.tquantity = Label(self.right, text='Quantity', font=('arial 18 bold'), bg='lightblue', fg='white')

self.tquantity.place(x=300, y=60)

self.tquantity.focus()

self.tamount = Label(self.right, text='Amount', font=('arial 18 bold'), bg='lightblue', fg='white')

self.tamount.place(x=500, y=60)

# entries.....

self.enterid = Label(self.left, text="Enter Product's ID", font=('arial 18 bold'), bg='white')

self.enterid.place(x=0, y=80)

self.enterid\_e = Entry(self.left, width=25, font=('arial 18 bold'), bg='lightblue')

self.enterid\_e.place(x=190, y=80)

self.enterid\_e.focus()

# buttons......

self.search\_btn = Button(self.left, text='Search', cursor='hand2', width=22, height=2, bg='lightblue', command=self.ajax)

self.search\_btn.place(x=350, y=120)

# fill later by function ajax...

self.productname = Label(self.left, text='', font=('arial 18 bold'), bg='white', fg='steelblue')

self.productname.place(x=0, y=250)

self.pprice = Label(self.left, text='', font=('arial 18 bold'), bg='white', fg='steelblue')

self.pprice.place(x=0, y=290)

# total label.....

self.total\_l = Label(self.right, text='', font=('arial 40 bold'), bg='lightblue', fg='white')

self.total\_l.place(x=0, y=600)

def logout(self):

for i in root.winfo\_children():

i.destroy()

Login(root)

def ajax(self):

self.get\_id = self.enterid\_e.get()

# get the products info with that id and fill in the labels above....

query = "SELECT \* FROM inventory WHERE id=?"

result = c.execute(query, (self.get\_id, ))

for self.r in result:

self.get\_id = self.r[0]

self.get\_name = self.r[1]

self.get\_price = self.r[4]

self.get\_stock = self.r[2]

self.productname.configure(text="Product's Name: " + str(self.get\_name))

self.pprice.configure(text="Price: INR. " + str(self.get\_price))

# create the quantity and the discount label

self.quantity\_l = Label(self.left, text='Enter Quantity:', font=('arial 18 bold'), bg='white')

self.quantity\_l.place(x=0, y=370)

self.quantity\_e = Entry(self.left, width=25, font=('arial 18 bold'), bg='lightblue')

self.quantity\_e.place(x=190, y=370)

self.quantity\_e.focus()

# discount ......

self.discount\_l = Label(self.left, text='Enter Discount:', font=('arial 18 bold'), bg='white')

self.discount\_l.place(x=0, y=410)

self.discount\_e = Entry(self.left, width=25, font=('arial 18 bold'), bg='lightblue')

self.discount\_e.place(x=190, y=410)

self.discount\_e.insert(END, 0)

# add to cart

self.add\_to\_cart\_btn = Button(self.left, text='Add to Cart', cursor="hand2", width=22, height=2, bg='lightblue', command=self.add\_cart)

self.add\_to\_cart\_btn.place(x=350, y=450)

# generate bill and change

self.change\_1 = Label(self.left, text='Cash:', font=('arial 18 bold'), bg='white')

self.change\_1.place(x=0, y=550)

self.change\_e = Entry(self.left, width=25, font='arial 18 bold', bg='lightblue')

self.change\_e.place(x=190, y=550)

# button change

self.change\_btn = Button(self.left, text='Calculate Change', cursor="hand2", width=22, height=2, bg='lightblue', command=self.change)

self.change\_btn.place(x=350, y=590)

# generate bill button

self.bill\_btn = Button(self.left, text='Generate Bill', cursor="hand2", font='arial 10 bold', width=80, height=2, bg='lightblue', fg='black', command=self.generate\_bill)

self.bill\_btn.place(x=30, y=640)

# update database....

self.update = Button(self.left, text='Update Products', cursor="hand2", font='arial 12 bold', width=25, height=2, bg='lightblue', fg='black', command=self.update\_products)

self.update.place(x=70, y=690)

# add to database......

self.add\_db = Button(self.left, text='Add Products', width=25, cursor="hand2", font='arial 12 bold', height=2, bg='lightblue', fg='black', command=self.add\_products)

self.add\_db.place(x=350, y=690)

def add\_cart(self):

# get the value from database

self.quantity\_value = int(self.quantity\_e.get())

if self.quantity\_value > int(self.get\_stock):

messagebox.showinfo('Error', 'Not that many products in our inventory')

else:

# calculate final price

self.final\_price = float(self.quantity\_value) \* float(self.get\_price) - float(str(self.discount\_e.get()))

products\_list.append(self.get\_name)

products\_price.append(self.final\_price)

products\_quantity.append(self.quantity\_value)

product\_id.append(self.get\_id)

# print(products\_list)

# print(products\_price)

# print(products\_quantity)

self.x\_index = 300

self.y\_index = 100

self.counter = 0

for self.p in products\_list:

self.tempname = Label(self.right, text=str(products\_list[self.counter]), font=('arial 18 bold'), bg='lightblue', fg='white')

self.tempname.place(x=0, y=self.y\_index)

label\_lists.append(self.tempname)

self.tempqt = Label(self.right, text=str(products\_quantity[self.counter]), font=('arial 18 bold'), bg='lightblue',

fg='white')

self.tempqt.place(x=300, y=self.y\_index)

label\_lists.append(self.tempqt)

self.tempprice = Label(self.right, text=str(products\_price[self.counter]), font='arial 18 bold', bg='lightblue',

fg='white')

self.tempprice.place(x=500, y=self.y\_index)

label\_lists.append(self.tempprice)

self.y\_index +=40

self.counter += 1

# total configuration

self.total\_l.configure(text='Total: INR. ' + str(sum(products\_price)))

# clear the window......

# self.quantity\_l.place\_forget()

# self.quantity\_e.place\_forget()

# self.discount\_l.place\_forget()

# self.discount\_e.place\_forget()

# self.productname.configure(text='')

# self.tempprice.configure(text='')

# self.add\_to\_cart\_btn.destroy()

# autofocus to enter id .....

self.enterid\_e.focus()

self.enterid\_e.delete(0, END)

def change(self):

# get the amount given by the customer and the amount generated by the the computer

self.amount\_given = float(self.change\_e.get())

self.our\_total = float(sum(products\_price))

self.to\_give = self.amount\_given - self.our\_total

# label change .....

self.c\_amount = Label(self.left, text="Change: INR. " + str(self.to\_give), font=('arial 18 bold'), fg='red', bg='white')

self.c\_amount.place(x=0, y=600)

def generate\_bill(self):

# create the bill before updating......

directory = "G:/mini3/Invoice/" + str(date) + "/"

if not os.path.exists(directory):

os.makedirs(directory)

# Templates for the bills

company = "\t\t\t\tStudent Choice Stationary.\n"

address = "\t\t\t\tShirpur, India.\n"

phone = "\t\t\t\tMobile No: 9977842528.\n"

sample = "\t\t\t\t\tInvoice.\n"

dt = "\t\t\t\t" + str(date)

table\_header = "\n\n\t\t=====================================================\n\t\t\tSN.\tProducts\t\tQty\t\tAmount\n\t\t====================================================="

final = company + address + phone + sample + dt + "\n" + table\_header

# open the file......

file\_name = str(directory) + str(random.randrange(5000, 10000)) + ".rtf"

f = open(file\_name, 'w')

f.write(final)

# fill the dynamic data.....

r = 1

i = 0

for t in products\_list:

f.write("\n\t\t\t" + str(r) + "\t" + str(products\_list[i] + ".......")[:7] + "\t\t" + str(products\_quantity[i]) + "\t\t" + str(products\_price[i]))

i += 1

r += 1

teller = "Prasoon"

f.write("\n\n\t\t\tTotal: INR. " + str(sum(products\_price)))

f.write("\n\n\t\t\tChange: INR. " + str(str(self.to\_give)))

f.write("\n\t\t\tThanks for shopping with us")

f.write("\n\t\t\tYou served by " + teller)

f.close()

# decrease the stock......

self.x = 0

initial = "SELECT \* FROM inventory WHERE id=?"

result = c.execute(initial, (product\_id[self.x], ))

for i in products\_list:

for r in result:

self.old\_stock = r[2]

self.new\_stock = int(self.old\_stock) - int(products\_quantity[self.x])

# updating the stock .......

sql = "UPDATE inventory SET stock=? WHERE id=?"

c.execute(sql, (self.new\_stock, product\_id[self.x]))

conn.commit()

# insert into trasanction table.......

sql2 = "INSERT INTO transactions (product\_name, quantity, amount, date)VALUES(?,?,?,?)"

trans = (products\_list[self.x], products\_quantity[self.x], products\_price[self.x], date)

c.execute(sql2, trans)

conn.commit()

self.x += 1

for a in label\_lists:

a.destroy()

del(products\_list[:])

del (product\_id[:])

del (products\_quantity[:])

del (products\_price[:])

self.total\_l.configure(text='')

self.enterid\_e.focus()

self.enterid\_e.delete(0, END)

self.c\_amount.configure(text='')

self.change\_e.delete(0, END)

messagebox.showinfo('Success', 'Done everything')

def add\_products(self):

for i in root.winfo\_children():

i.destroy()

Database(root)

def update\_products(self):

for i in root.winfo\_children():

i.destroy()

UpdateProducts(root)

# update class......

result = c.execute("SELECT Max(id) FROM inventory")

for r in result:

id = r[0]

class UpdateProducts:

def \_\_init\_\_(self, master):

self.master = master

self.heading = Label(master, text='Update Products', font='arial 40 bold', fg='steelblue', bg='#d9d9d9')

self.heading.place(x=400, y=0)

# label and entry for id

self.Id\_l = Label(master, text="Enter Id", font='arial 18 bold', bg='#d9d9d9')

self.Id\_l.place(x=0, y=70)

self.Id\_e = Entry(master, font='arial 18 bold', width=10, bg='white')

self.Id\_e.place(x=380, y=70)

self.btn\_search = Button(master, text='Search', width=15, height=2, bg='lightblue', command=self.search)

self.btn\_search.place(x=550, y=70)

# labels and entries for the window

self.name\_l = Label(master, text='Enter Product Name', font='arial 18 bold', bg='#d9d9d9')

self.name\_l.place(x=0, y=120)

self.stock\_l = Label(master, text='Enter Stock', font='arial 18 bold', bg='#d9d9d9')

self.stock\_l.place(x=0, y=170)

self.cp\_l = Label(master, text='Enter Cost Price', font='arial 18 bold', bg='#d9d9d9')

self.cp\_l.place(x=0, y=220)

self.sp\_l = Label(master, text='Enter Selling Price', font='arial 18 bold', bg='#d9d9d9')

self.sp\_l.place(x=0, y=270)

self.totalcp\_l = Label(master, text='Enter Total Cost Price', font='arial 18 bold', bg='#d9d9d9')

self.totalcp\_l.place(x=0, y=320)

self.totalsp\_l = Label(master, text='Enter Total Selling Price', font='arial 18 bold', bg='#d9d9d9')

self.totalsp\_l.place(x=0, y=370)

self.vendor\_l = Label(master, text='Enter Vendor Name', font='arial 18 bold', bg='#d9d9d9')

self.vendor\_l.place(x=0, y=420)

self.vendor\_phone\_l = Label(master, text='Enter Vendor Phone Number', font='arial 18 bold', bg='#d9d9d9')

self.vendor\_phone\_l.place(x=0, y=470)

# entries for the label....

self.name\_e = Entry(master, width=25, font='arial 18 bold')

self.name\_e.place(x=380, y=120)

self.stock\_e = Entry(master, width=25, font='arial 18 bold')

self.stock\_e.place(x=380, y=170)

self.cp\_e = Entry(master, width=25, font='arial 18 bold')

self.cp\_e.place(x=380, y=220)

self.sp\_e = Entry(master, width=25, font='arial 18 bold')

self.sp\_e.place(x=380, y=270)

self.totalcp\_e = Entry(master, width=25, font='arial 18 bold')

self.totalcp\_e.place(x=380, y=320)

self.totalsp\_e = Entry(master, width=25, font='arial 18 bold')

self.totalsp\_e.place(x=380, y=370)

self.vendor\_e = Entry(master, width=25, font='arial 18 bold')

self.vendor\_e.place(x=380, y=420)

self.vendor\_phone\_e = Entry(master, width=25, font='arial 18 bold')

self.vendor\_phone\_e.place(x=380, y=470)

# button to add to database

self.btn\_add = Button(master, text='Update Database', width=25, height=2, bg='steelblue', fg='white', cursor="hand2", command=self.update)

self.btn\_add.place(x=520, y=520)

# button to go back to main application..........

def back\_to\_main():

for i in root.winfo\_children():

i.destroy()

Application(root)

self.btn\_add = Button(master, text='Back', width=50, height=2, bg='#d9d9d9', fg='#4d4d1a', cursor="hand2", command=back\_to\_main)

self.btn\_add.place(x=350, y=620)

# text box

self.tbox = Text(master, width=60, height=17)

self.tbox.place(x=750, y=70)

self.tbox.insert(END, 'ID has reached up to:' + str(id))

def search(self):

sql = "SELECT \* FROM inventory WHERE id=?"

result = c.execute(sql, (self.Id\_e.get(),))

for r in result:

self.n1 = r[1] # name

self.n2 = r[2] # stock

self.n3 = r[3] # cp

self.n4 = r[4] # sp

self.n5 = r[5] # totalcp

self.n6 = r[6] # totalsp

self.n7 = r[7] # assumed\_profit

self.n8 = r[8] # vendor

self.n9 = r[9] # vendor\_phone

conn.commit()

# insert into the entries to update

self.name\_e.delete(0, END)

self.name\_e.insert(0, str(self.n1))

self.stock\_e.delete(0, END)

self.stock\_e.insert(0, str(self.n2))

self.cp\_e.delete(0, END)

self.cp\_e.insert(0, str(self.n3))

self.sp\_e.delete(0, END)

self.sp\_e.insert(0, str(self.n4))

self.totalcp\_e.delete(0, END)

self.totalcp\_e.insert(0, str(self.n5))

self.totalsp\_e.delete(0, END)

self.totalsp\_e.insert(0, str(self.n6))

self.vendor\_e.delete(0, END)

self.vendor\_e.insert(0, str(self.n8))

self.vendor\_phone\_e.delete(0, END)

self.vendor\_phone\_e.insert(0, str(self.n9))

def update(self):

# get the update values

self.u1 = self.name\_e.get()

self.u2 = self.stock\_e.get()

self.u3 = self.cp\_e.get()

self.u4 = self.sp\_e.get()

self.u5 = self.totalcp\_e.get()

self.u6 = self.totalsp\_e.get()

self.u7 = self.vendor\_e.get()

self.u8 = self.vendor\_phone\_e.get()

query = "UPDATE inventory SET name=?, stock=?, cp=?, sp=?, totalcp=?, totalsp=?, vendor=?, vendor\_phoneno=? WHERE id=?"

values = (self.u1, self.u2, self.u3, self.u4, self.u5, self.u6, self.u7, self.u8, self.Id\_e.get())

c.execute(query, values)

conn.commit()

messagebox.showinfo('success', 'updated database')

# add to database class ........

result = c.execute("SELECT Max(id) FROM inventory")

for r in result:

id = r[0]

class Database:

def \_\_init\_\_(self, master):

self.master = master

self.heading = Label(master, text='Add to Database', font=('arial 40 bold'), fg='steelblue', bg='#d9d9d9')

self.heading.place(x=400, y=0)

# labels and entries for the window

self.name\_l = Label(master, text='Enter Product Name', font='arial 18 bold', bg='#d9d9d9')

self.name\_l.place(x=0, y=70)

self.stock\_l = Label(master, text='Enter Stock', font='arial 18 bold', bg='#d9d9d9')

self.stock\_l.place(x=0, y=120)

self.cp\_l = Label(master, text='Enter Cost Price', font='arial 18 bold',bg='#d9d9d9')

self.cp\_l.place(x=0, y=170)

self.sp\_l = Label(master, text='Enter Selling Price', font='arial 18 bold',bg='#d9d9d9')

self.sp\_l.place(x=0, y=220)

self.vendor\_l = Label(master, text='Enter Vendor Name', font='arial 18 bold', bg='#d9d9d9')

self.vendor\_l.place(x=0, y=270)

self.vendor\_phone\_l = Label(master, text='Enter Vendor Phone Number', font='arial 18 bold', bg='#d9d9d9')

self.vendor\_phone\_l.place(x=0, y=320)

self.id\_l = Label(master, text='Enter ID', font='arial 18 bold', bg='#d9d9d9')

self.id\_l.place(x=0, y=370)

# entries for the label....

self.name\_e = Entry(master, width=25, font='arial 18 bold')

self.name\_e.place(x=380, y=70)

self.stock\_e = Entry(master, width=25, font='arial 18 bold')

self.stock\_e.place(x=380, y=120)

self.cp\_e = Entry(master, width=25, font='arial 18 bold')

self.cp\_e.place(x=380, y=170)

self.sp\_e = Entry(master, width=25, font='arial 18 bold')

self.sp\_e.place(x=380, y=220)

self.vendor\_e = Entry(master, width=25, font='arial 18 bold')

self.vendor\_e.place(x=380, y=270)

self.vendor\_phone\_e = Entry(master, width=25, font='arial 18 bold')

self.vendor\_phone\_e.place(x=380, y=320)

self.id\_e = Entry(master, width=25, font=('arial 18 bold'))

self.id\_e.place(x=380, y=370)

# button to add to database

self.btn\_add = Button(master, text='Add To Database', width=25, cursor="hand2", height=2, bg='steelblue', fg='white', command=self.get\_items)

self.btn\_add.place(x=520, y=420)

self.btn\_clear = Button(master, text='Clear All Fields', width=18, height=2, bg='lightgreen',

fg='white', cursor="hand2", command=self.clear\_all)

self.btn\_clear.place(x=350, y=420)

# button to go back to main application..........

def back\_to\_main():

for i in root.winfo\_children():

i.destroy()

Application(root)

self.btn\_add = Button(master, text='Back', width=50, height=2, bg='#d9d9d9', fg='#4d4d1a', cursor="hand2", command=back\_to\_main)

self.btn\_add.place(x=350, y=520)

# text box

self.tbox = Text(master, width=60, height=20)

self.tbox.place(x=750, y=70)

self.tbox.insert(END, 'ID has reached up to:' + str(id))

def get\_items(self):

self.name = self.name\_e.get()

self.stock = self.stock\_e.get()

self.cp = self.cp\_e.get()

self.sp = self.sp\_e.get()

self.vendor = self.vendor\_e.get()

self.vendor\_phone = self.vendor\_phone\_e.get()

# dynamic entries...

self.totalcp = float(self.cp) \* float(self.stock)

self.totalsp = float(self.sp) \* float(self.stock)

self.assumed\_profit = float(self.totalsp-self.totalcp)

if self.name == ' ' or self.stock == ' ' or self.cp == ' ' or self.sp == ' ':

messagebox.showinfo('Error', 'Please fill all entries')

else:

sql = "INSERT INTO inventory (name, stock, cp, sp, totalcp, totalsp, assumed\_profit, vendor, vendor\_phoneno)VALUES(?,?,?,?,?,?,?,?,?)"

parameters = (self.name, self.stock, self.cp, self.sp, self.totalcp, self.totalsp, self.assumed\_profit, self.vendor, self.vendor\_phone)

c.execute(sql, parameters)

conn.commit()

# textbox insert

self.tbox.insert(END, "\n\nInserted\t" + str(self.name) + "\tinto the database with code.\t" + str(self.id\_e.get()))

messagebox.showinfo('success!!', 'Successfully added to the database')

def clear\_all(self):

num = id + 1

self.name\_e.delete(0, END)

self.stock\_e.delete(0, END)

self.cp\_e.delete(0, END)

self.sp\_e.delete(0, END)

self.vendor\_e.delete(0, END)

self.vendor\_phone\_e.delete(0, END)

self.id\_e.delete(0, END)

'''This class configures and populates the toplevel window.

top is the toplevel containing window.'''

\_bgcolor = '#d9d9d9' # X11 color: 'gray85'

\_fgcolor = '#000000' # X11 color: 'black'

\_compcolor = '#d9d9d9' # X11 color: 'gray85'

\_ana1color = '#d9d9d9' # X11 color: 'gray85'

\_ana2color = '#d9d9d9' # X11 color: 'gray85'

font10 = "-family {Courier New} -size 10 -weight normal -slant" \

" roman -underline 0 -overstrike 0"

font11 = "-family {Kristen ITC} -size 34 -weight normal -slant" \

" roman -underline 0 -overstrike 0"

font12 = "-family {Segoe UI} -size 18 -weight normal -slant " \

"italic -underline 0 -overstrike 0"

font13 = "-family {Segoe UI} -size 14 -weight normal -slant " \

"roman -underline 0 -overstrike 0"

font17 = "-family {Gill Sans MT Condensed} -size 14 -weight " \

"normal -slant italic -underline 0 -overstrike 0"

font18 = "-family {Segoe UI} -size 9 -weight normal -slant " \

"italic -underline 0 -overstrike 0"

class Login:

def \_\_init\_\_(self, master):

self.Frame1 = Frame(master)

self.Frame1.place(relx=-0.02, rely=-0.02, relheight=0.21, relwidth=1.04)

self.Frame1.configure(relief=FLAT, borderwidth="2", background="#0066ff", width=625)

self.Label1 = Label(self.Frame1)

self.Label1.place(relx=0.23, rely=0.38, height=41, width=700)

self.Label1.configure(background="#0066ff", borderwidth="0", disabledforeground="#a3a3a3", font=font11, foreground="#ffffff", text='''Students Choice Stationary''', width=200)

self.loginLabel = Label(master)

self.loginLabel.place(relx=0.43, rely=0.30, height=31, width=94)

self.loginLabel.configure(background="#d9d9d9", disabledforeground="#a3a3a3", font='arial 18 bold', foreground="#000000", width=94)

self.loginLabel.configure(text='''Log In''')

self.user\_l = Label(master)

self.user\_l.place(relx=0.25, rely=0.40, height=31, width=144)

self.user\_l.configure(activebackground="#f9f9f9", activeforeground="black", background="#d9d9d9", disabledforeground="#a3a3a3", font='arial 18 bold')

self.user\_l.configure(foreground="#000000", highlightbackground="#d9d9d9", highlightcolor="black")

self.user\_l.configure(text='''Username''')

self.txtuser\_e = Entry(master)

self.txtuser\_e.place(relx=0.4, rely=0.40, height=30, relwidth=0.15)

self.txtuser\_e.configure(background="white", borderwidth="0", disabledforeground="#a3a3a3", font='arial 18 bold', foreground="#000000", highlightbackground="#d9d9d9")

self.txtuser\_e.configure(highlightcolor="black", insertbackground="black", selectbackground="#c4c4c4", selectforeground="black")

self.pass\_l = Label(master)

self.pass\_l.place(relx=0.25, rely=0.50, height=31, width=144)

self.pass\_l.configure(activebackground="#f9f9f9", activeforeground="black", background="#d9d9d9", disabledforeground="#a3a3a3", font='arial 18 bold')

self.pass\_l.configure(foreground="#000000", highlightbackground="#d9d9d9", highlightcolor="black")

self.pass\_l.configure(text='''Password''')

self.txtpass\_e = Entry(master, show='\*')

self.txtpass\_e.place(relx=0.4, rely=0.50, height=30, relwidth=0.15)

self.txtpass\_e.configure(background="white", borderwidth="0", disabledforeground="#a3a3a3", font='arial 18 bold', foreground="#000000", highlightbackground="#d9d9d9")

self.txtpass\_e.configure(highlightcolor="black", insertbackground="black", selectbackground="#c4c4c4", selectforeground="black")

self.BtnLogin = Button(master)

self.BtnLogin.place(relx=0.42, rely=0.60, height=34, width=150)

self.BtnLogin.configure(activebackground="#d9d9d9", activeforeground="#000000", font='arial 10 bold', background="#d9d9d9", cursor="hand2", disabledforeground="#a3a3a3")

self.BtnLogin.configure(foreground="black", highlightbackground="#d9d9d9", highlightcolor="black", pady="0", width=100)

self.BtnLogin.configure(text='''Login''', command=self.login)

def login(self):

if self.txtuser\_e.get() == username and self.txtpass\_e.get() == 'admin':

for i in root.winfo\_children():

i.destroy()

Application(root)

else:

messagebox.showerror('Invalid credentials', 'Try again with the correct credentials!!!!')

root = Tk()

Login(root)

root.title("Students Choice Stationary")

root.iconbitmap(r'studentschoice.icon')

root.configure(background="#d9d9d9")

root.resizable(False, False)

root.geometry("1366x768+0+0")

root.mainloop()

**Output Procedure:**

