Odd Semester (2023)

Assignment Cover Letter

(Individual Work)

Student Information:		**Surname**	**Given Names**	**Student ID Number**
	1.	**Lucianto**	**William**	**2301890390**

Course Code	**: COMP6502**	**Course Name**	**: Introduction to Programming**
Class	**: L1AC**	**Name of Lecturer(s)**	**:**Ida Bagus Kerthyayana
Major	**: CS**		
Title of Assignment (if any)	: Inventory management		
Type of Assignment	**: Final Project**		
Submission Pattern			
Due Date	**: 14-01-20**	**Submission Date**	**: 14-01-20**

Signature of Student: (Name of Student)
1. William Lucianto Santoso

"inventory management"

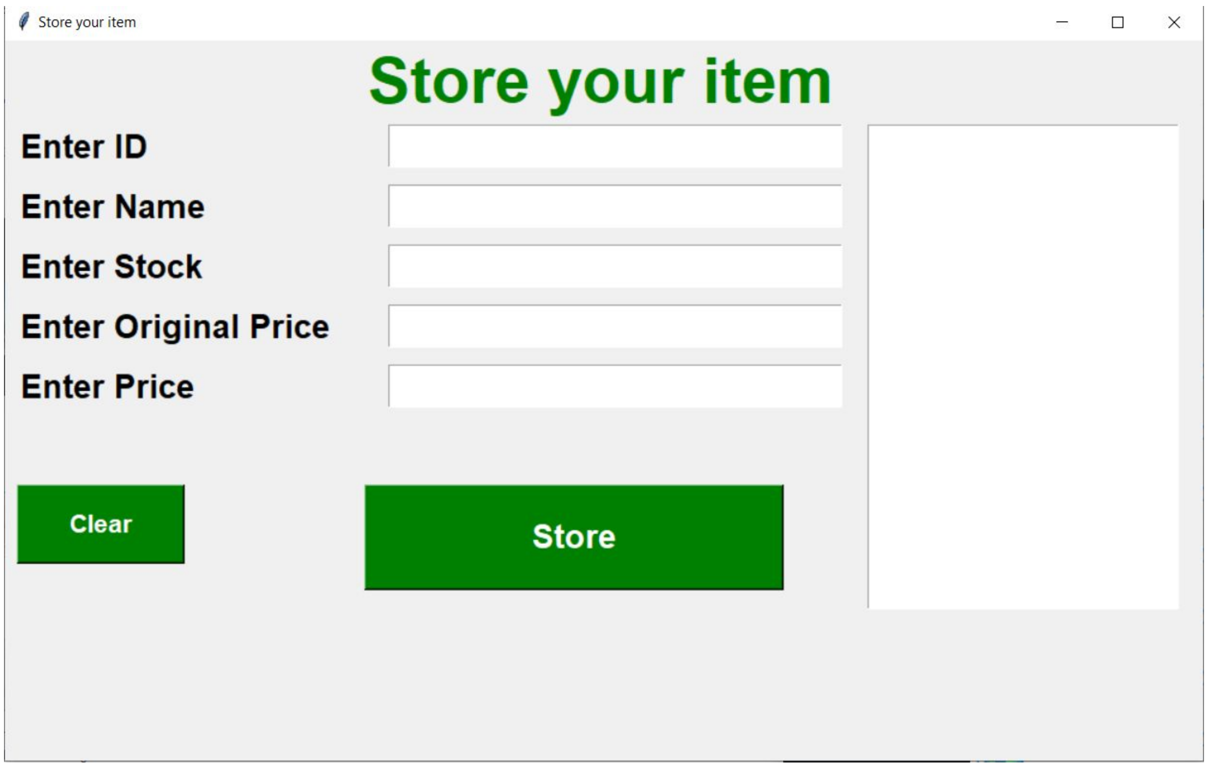
Name :William Lucianto Santoso

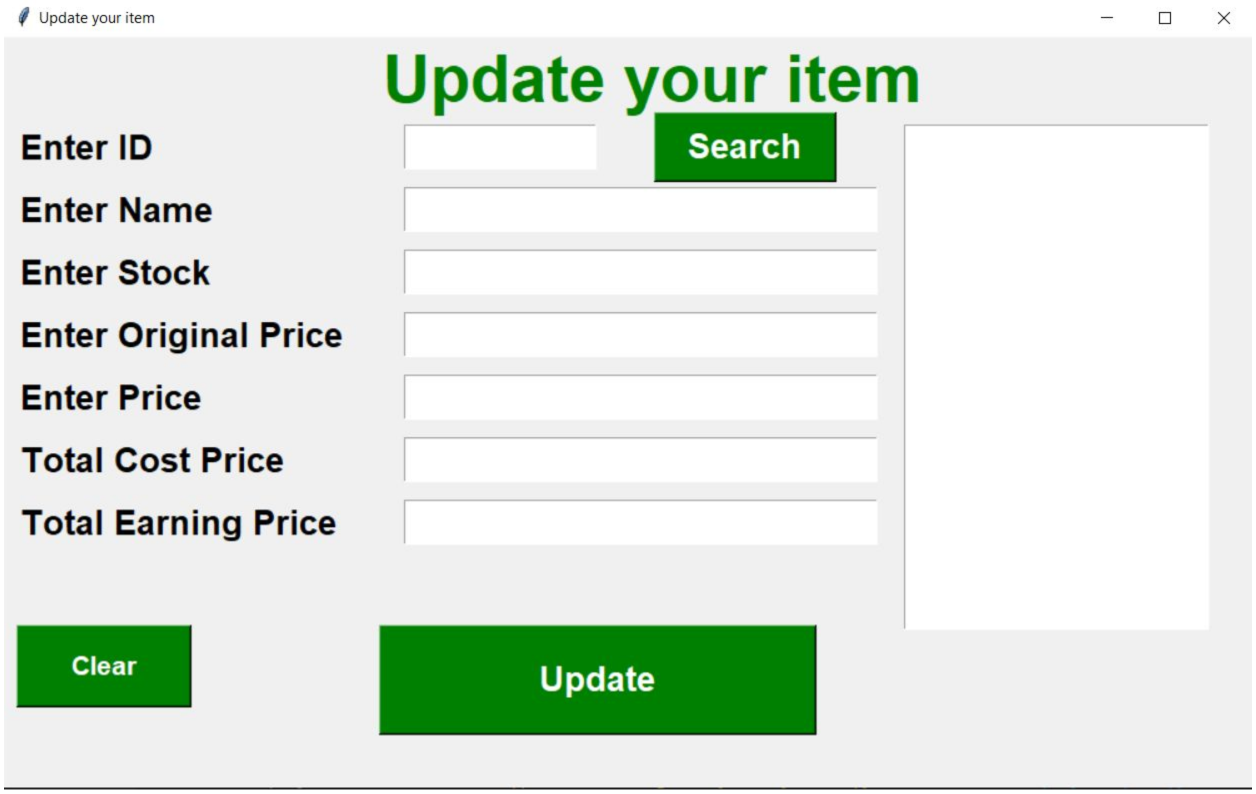
ID : 2301890390

I. Description

The function of this program:

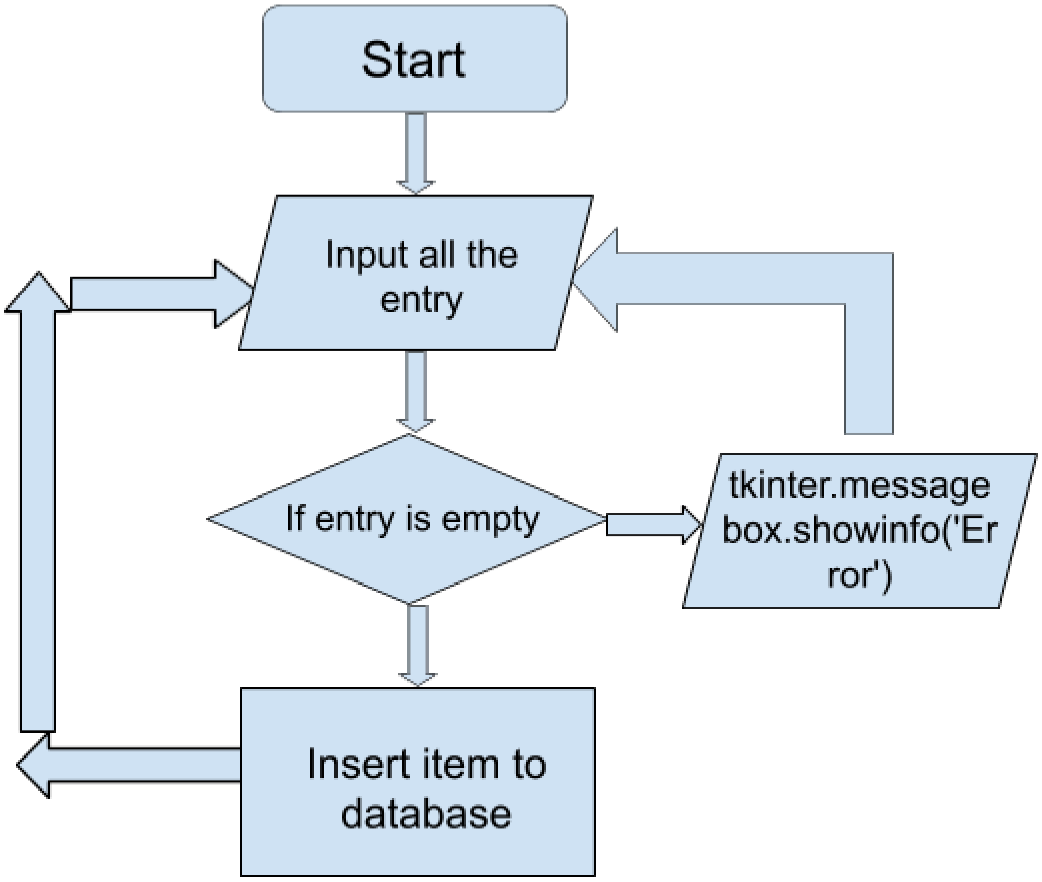
The purpose of this program is to store and update item into database using sqlite3 so the user can easily track their items and they can search the item and modify it with this program. This program use Graphic User Interface by tkinter for a better UI & UX

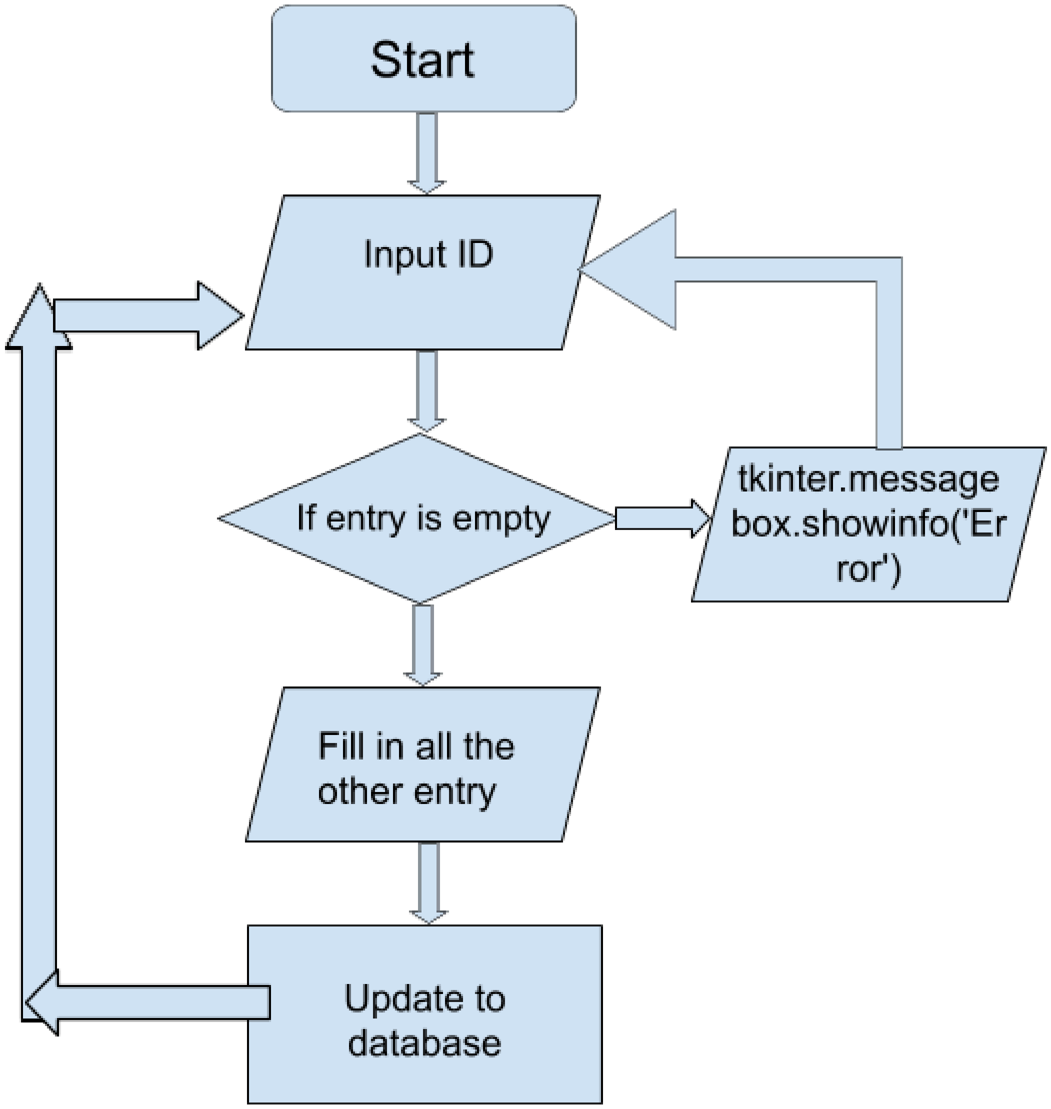

Store your item
Store your item
Enter ID
Enter Name
Enter Stock
Enter Original Price
Enter Price
Clear
Store


Update your item
Update your item
Enter ID
Search
Enter Name
Enter Stock
Enter Original Price
Enter Price
Total Cost Price
Total Earning Price
Clear
Update

II.a. Design/Plan

Project's Flow Chart




Start
Input ID
If entry is empty
tkinter.message box.showinfo('Error')
Fill in all the other entry
Update to database

I.b. Explanation of Each Function Inside the Class

- **delete_entry (self, *args, **kwargs) :**
    - delete all Entry to make all the Entries empty
- **store_items (self, *args, **kwargs) :**
    - use get() to take the value from the “clothe” table
    - if there is an empty entry, show error messages by using if and else
    - if entry is filled, calculate the total cost by (cost x stock), total price (price x stock), assumed profit (total earning - total cost)
    - use sql command “INSERT INTO clothe” to input all the data into the table
    - use c.execute() to perform the sql command
    - commit change by using conn.commit()
    - add message box to show that the storing process is completed
    - input a “Added {} into your database” ({} = item you want to store) into your entry log
- **search (self, *args, **kwargs) :**
    - use try and except to make sure there are no error by putting error message box in the except
    - use sql command “SELECT * FROM clothe WHERE id=?
    - use c.execute to use the sql command
    - since there are 7 key in the table, we make a 7 new variable with loop where each variable have the values for each keys
    - commit change by using conn.commit()

    - use delete to remove all the entries if there are any, then insert the new variable to get the values in the entry

- **update (self, *args, **kwargs):**
    - use of try and except to make sure there are no errors
    - make a new variable to keep the updated values
    - put the updated values to the new variable by using get()
    - use sql command = “UPDATE clothe SET name=?, id=?, and so on
    - c.execute to use the sql command
    - conn.commit() to commit the change in table
    - insert message in the entry log that the update is successful
    - insert message showinfo to make a pop up to show success

III.a. Lessons that Have Been Learned

```
from tkinter import *
import sqlite3
import tkinter.messagebox
from fungsi import *
```

- I learned how to import all by using from …. import *

```
root = Tk()
b = Database(root)

root.geometry('1000x600')
root.title("Update your item")
root.mainloop()
```

- I have learned the basics of tkinter where you create root window, create an instance for it, create the size & title before you use mainloop() to loop & spawn you the window.

```
conn = sqlite3.connect("D:\Final project\Database\stock.db")
c = conn.cursor()
```

- learned the basic of sqlite where you connect the file into the database file

```
class Database:
        def __init__(self, master, *args, **kwargs):

                self.master = master
                self.heading = Label(master, text="Store your item",
font=('arial 40 bold'), fg='green')
                self.heading.place(x=300, y=0)
```

- the basic of tkinter where you can make label, entry, checkbox, button, and many more by making the variable and then use the (x,y) to put it wherever you want

```
                              sql = 'INSERT INTO clothe (id, name, stock,
originalPrice, price, totalCost, totalEarning, assumedProfit)
VALUES(?,?,?,?,?,?,?,?)'
                              c.execute(sql, (self.id, self.name, self.stock,
self.originalPrice, self.price, self.totalCost, self.totalEarning,
self.assumedProfit))
                              conn.commit()
```

- the use of sql command such as “INSERT, UPDATE, INTO, SELECT, FROM, WHERE, and many more)
- also the use of c.execute to use the sql command
- and the use of conn.commit() to commit the change of the database

```
tkinter.messagebox.showinfo("SUCCESS", 'Your item have been stored')
```

- how to make a pop up message box

III.b. Problem that Have Been Overcome

Creating this is program is actually not that hard after a lot of research regarding tkinter and sqlite3 since you only need the basic to make this program, but what i found hard to do is the time since I don’t really have much time because after vacation turned out there is flood in Jakarta so my house don’t have electricity. I really wanted to add one more function which is a cashier but my family need to go to Semarang at Friday 10th January until Sunday 12th January so I don’t have enough time to make it and ended up stopping midway. I also tried to make the executable files for both program with pyinstaller, but the Database.py can’t be open somehow and i try fixing it by searching around people with the same problem with no luck.

Resources :

-https://www.youtube.com/watch?v=Wgjja1-K_kQ&list=PLeyK9Dw9ShReNENDOoG5r133npUKF5IhD

V. Source Code

database.py

```
from tkinter import *
import sqlite3
import tkinter.messagebox

conn = sqlite3.connect("D:\Final project\Database\stock.db")
c = conn.cursor()

class Database:
    def __init__(self, master, *args, **kwargs):

        self.master = master
        self.heading = Label(master, text="Store your item", font=('arial 40 bold'), fg='green')
        self.heading.place(x=300, y=0)

# Labels

        self.id1 = Label(master, text="Enter ID", font=('arial 20 bold'))
        self.id1.place(x=10, y=70)

        self.name1 = Label(master, text="Enter Name", font=('arial 20 bold'))
        self.name1.place(x=10, y=120)

        self.stock1 = Label(master, text="Enter Stock", font=('arial 20 bold'))
        self.stock1.place(x=10, y=170)

        self.originalPrice1 = Label(master, text="Enter Original Price", font=('arial 20 bold'))
        self.originalPrice1.place(x=10, y=220)

        self.price1 = Label(master, text="Enter Price", font=('arial 20 bold'))
        self.price1.place(x=10, y=270)


#ENTRY

        self.id_e = Entry(master, width=25,font=('arial 20 bold'))
        self.id_e.place(x=320,y=70)

        self.name_e = Entry(master, width=25,font=('arial 20 bold'))
        self.name_e.place(x=320,y=120)

        self.stock_e = Entry(master, width=25,font=('arial 20 bold'))
        self.stock_e.place(x=320,y=170)

        self.originalPrice_e = Entry(master, width=25,font=('arial 20 bold'))
```

```
            self.originalPrice_e.place(x=320,y=220)

            self.price_e = Entry(master, width=25,font=('arial 20 bold'))
            self.price_e.place(x=320,y=270)


# BUTTON

            self.btn_store = Button(master, text="Store",font=('arial 20 bold'), width=20, height=2,
bg='green', fg='white', command=self.store_items)
            self.btn_store.place(x=300,y=370)
            self.btn_clear = Button(master, text="Clear", font=('arial 16 bold'), width=10, height=2,
bg='green', fg='white', command=self.delete_entry)
            self.btn_clear.place(x=10,y=370)

#ENTRY LOG

            self.log = Text(master, width=32,height=25)
            self.log.place(x=720, y=70)
#FUNCTION
      def delete_entry(self, *args, **kwargs):
            self.id_e.delete(0, END)
            self.name_e.delete(0, END)
            self.stock_e.delete(0, END)
            self.originalPrice_e.delete(0, END)
            self.price_e.delete(0, END)

      def store_items(self, *args, **kwargs):
            self.id = self.id_e.get()
            self.name = self.name_e.get()
            self.stock = self.stock_e.get()
            self.originalPrice = self.originalPrice_e.get()
            self.price = self.price_e.get()


            if self.id == '' or self.name == '' or self.stock == '' or self.originalPrice == '' or self.price == '':
                  tkinter.messagebox.showinfo('Error', 'Please fill all the entries')
            else:
                  self.totalCost = float(self.originalPrice) * float(self.stock)
                  self.totalEarning = float(self.price) * float(self.stock)
                  self.assumedProfit = float(self.totalEarning - self.totalCost)
                  sql = 'INSERT INTO clothe (id, name, stock, originalPrice, price, totalCost, totalEarning,
assumedProfit) VALUES(?,?,?,?,?,?,?,?)'
                  c.execute(sql, (self.id, self.name, self.stock, self.originalPrice, self.price, self.totalCost,
self.totalEarning, self.assumedProfit))
                  conn.commit()
                  self.log.insert(END, 'Added ' + str(self.name) + ' into your database\n')
                  tkinter.messagebox.showinfo("SUCCESS", 'Your item have been stored')
```

```
root = Tk()
b = Database(root)

root.geometry('1000x600')
root.title("Store your item")
root.mainloop()
```

update.py

```
from tkinter import *
import sqlite3
import tkinter.messagebox

conn = sqlite3.connect("D:\Final project\Database\stock.db")
c = conn.cursor()

class Database:
        def __init__(self, master, *args, **kwargs):

                self.master = master
                self.heading = Label(master, text="Update your item", font=('arial 40 bold'),
fg='green')
                self.heading.place(x=300, y=0)
```

```
#Label & Entry
        self.id1 = Label(master, text="Enter ID", font=('arial 20 bold'))
        self.id1.place(x=10, y=70)

        self.id_entry = Entry(master, font=('arial 20 bold'), width=10)
        self.id_entry.place(x=320,y=70)

        self.id_btn = Button(master, text='Search', font=('arial 20 bold'),width=8, height=1,
bg='green', fg='white', command=self.search)
        self.id_btn.place(x=520,y=60)
#Labels
        self.name1 = Label(master, text="Enter Name", font=('arial 20 bold'))
        self.name1.place(x=10, y=120)

        self.stock1 = Label(master, text="Enter Stock", font=('arial 20 bold'))
        self.stock1.place(x=10, y=170)

        self.originalPrice1 = Label(master, text="Enter Original Price", font=('arial 20 bold'))
        self.originalPrice1.place(x=10, y=220)

        self.price1 = Label(master, text="Enter Price", font=('arial 20 bold'))
        self.price1.place(x=10, y=270)

        self.totalCost1 = Label(master, text="Total Cost Price", font=('arial 20 bold'))
        self.totalCost1.place(x=10, y=320)

        self.totalEarning1 = Label(master, text="Total Earning Price", font=('arial 20 bold'))
        self.totalEarning1.place(x=10, y=370)


#ENTRY

        self.name_e = Entry(master, width=25,font=('arial 20 bold'))
        self.name_e.place(x=320,y=120)

        self.stock_e = Entry(master, width=25,font=('arial 20 bold'))
        self.stock_e.place(x=320,y=170)

        self.originalPrice_e = Entry(master, width=25,font=('arial 20 bold'))
        self.originalPrice_e.place(x=320,y=220)

        self.price_e = Entry(master, width=25,font=('arial 20 bold'))
        self.price_e.place(x=320,y=270)

        self.totalCost_e = Entry(master, width=25,font=('arial 20 bold'))
        self.totalCost_e.place(x=320,y=320)
        self.totalEarning_e = Entry(master, width=25,font=('arial 20 bold'))
        self.totalEarning_e.place(x=320,y=370)



#BUTTON
        self.btn_store = Button(master, text="Update",font=('arial 20 bold'), width=20,
height=2, bg='green', fg='white', command=self.update)
        self.btn_store.place(x=300,y=470)
        self.btn_clear = Button(master, text="Clear", font=('arial 16 bold'), width=10,
height=2, bg='green', fg='white', command=self.delete_entry)
        self.btn_clear.place(x=10,y=470)
#ENTRY LOG
        self.log = Text(master, width=30,height=25)
        self.log.place(x=720, y=70)
#FUNCTION
    def delete_entry(self, *args, **kwargs):
        self.id_entry.delete(0, END)
        self.name_e.delete(0, END)
        self.stock_e.delete(0, END)
        self.originalPrice_e.delete(0, END)
        self.price_e.delete(0, END)
        self.totalCost_e.delete(0, END)
        self.totalEarning_e.delete(0, END)

    def search(self, *args, **kwargs):
        try:
            sql = "SELECT * FROM clothe WHERE id=?"
            result = c.execute(sql, (self.id_entry.get(), ))
            for r in result:
                self.n1 = r[1]
                self.n2 = r[2]
                self.n3 = r[3]
                self.n4 = r[4]
                self.n5 = r[5]
                self.n6 = r[6]
                self.n7 = r[7]
            conn.commit()
            #insert 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.originalPrice_e.delete(0, END)
            self.originalPrice_e.insert(0, str(self.n3))

            self.price_e.delete(0, END)
```

```
                self.price_e.insert(0, str(self.n4))

                self.price_e.delete(0, END)
                self.price_e.insert(0, str(self.n4))

                self.price_e.delete(0, END)
                self.price_e.insert(0, str(self.n4))

                self.totalCost_e.delete(0, END)
                self.totalCost_e.insert(0, str(self.n5))

                self.totalEarning_e.delete(0, END)
                self.totalEarning_e.insert(0, str(self.n6))
            except:
                tkinter.messagebox.showinfo('Error', 'Please input correctly')

    def update(self, *args, **kwargs):
            try:
                #get all the updated value
                self.u_id = self.id_entry.get()
                self.u1 = self.name_e.get()
                self.u2 = self.stock_e.get()
                self.u3 = self.originalPrice_e.get()
                self.u4 = self.price_e.get()
                self.u5 = float(self.u3) * float(self.u2)
                self.u6 = float(self.u4) * float(self.u2)
                self.u7 = float(self.u6) - float(self.u5)



                query = "UPDATE clothe SET name=?, stock=?, originalPrice=?, price=?,
totalCost=?, totalEarning=?, assumedProfit=? WHERE id=?"
                c.execute(query, (self.u1, self.u2, self.u3, self.u4, self.u5, self.u6,self.u7,
self.u_id))
                conn.commit()
                self.log.insert(END, 'Your item have been updated\n')
                tkinter.messagebox.showinfo("Success", "Your item have been updated")
            except:
                tkinter.messagebox.showinfo('Error', 'Please input correctly')


root = Tk()
b = Database(root)

root.geometry('1000x600')
root.title("Update your item")
root.mainloop()
```