**INTRODUCTION**

**Inventory management**  is a [software system](https://en.wikipedia.org/wiki/Software_system) for tracking [inventory](https://en.wikipedia.org/wiki/Inventory) levels, orders, [sales](https://en.wikipedia.org/wiki/Sales) and deliveries. It can also be used in the [manufacturing](https://en.wikipedia.org/wiki/Manufacturing) industry to create a [work order](https://en.wikipedia.org/wiki/Work_order), [bill of materials](https://en.wikipedia.org/wiki/Bill_of_materials) and other production-related documents. Companies use inventory management software to avoid product [overstock](https://en.wikipedia.org/wiki/Overstock) and outages. It is a tool for organizing inventory [data](https://en.wikipedia.org/wiki/Data) that before was generally stored in hard-copy form or in [spreadsheets](https://en.wikipedia.org/wiki/Spreadsheet). Effective maintenance of the inventory costs is very important. Inventory management is one of the crucial tasks that the industries need to handle at times. Businesses ranging from small to large businesses must manage, control and track the inventory from time to time and from anywhere. Control and management of the inventories may be the small or large businesses are very important. The inventory control management database system is actually the documenting the details of the inventories present in the industries to reach the goal. The inventory control management database system should be designed in such a way that you should be able to obtain the low raw material prices.

With the use of Tkinter and sqlite3 library package, we managed to make this project a success. Here, sqlite3 represents the database which will help in executing all the SQL commands. Alongside, tkinter helps creating the GUI (graphical user interface) applications.

The inventory management ensures that the company always has the required materials and products in hand while keeping the cost as low as possible. Inventory Management refers to the process of supervising and controlling the stock items of a company. Typically, Inventory Management Systems are used by firms that either sell a product or manufacture a product for the purpose of accounting all the tangible goods that allow for a sale of a finished product, or parts for making a product. This inventory management system can be used to store the details of the inventory, update the inventory based on the sale details, generate receipts for sales, generate sales and inventory reports periodically. This inventory management software has one module, Admin. Admin has the authority to add, update and delete an inventory. This inventory management software also has its own intelligently managed support system. This intelligent support system allows admin to view and manage various inventories.

This project will give the project synopsis of the inventory management system. The process of maintaining the information of the inventories at one stretch is very difficult. If you want the information once obtained about the inventory, it should be made available even the next time when you need that information. At this situation the inventory management system comes into picture. The inventory management system can have a database that can maintain proper management of variety of items. It can also increase the inventory turnover ad also optimize the sales stock levels.

**OBJECTIVE**

The main objective of the inventory management system is to manage details of user, category, goods, manager and such information. The process of maintaining the information of the inventories at one stretch is very difficult. If you want the information once obtained about the inventory, it should be made available even the next time when you need that information. At this situation the inventory management system comes into picture.

The inventory management system can have a database that can maintain proper management of variety of items. It can also increase the inventory turnover ad also optimize the sales stock levels. It manages the information about them. His project is totally built at administrative end and thus only the administrator is guaranteed the access. The purpose of the project is to build an application program to reduce the manual work for managing the abovementioned factors and tracking down their details. The other goals for his project will be providing user-friendliness, duplicity, security and fastness to boot up the features of the inventory.

**MODULES**

Simple Inventory System project is written in Python using Tkinter for GUI. This is a simple GUI based project which is easy to understand and use. After analysing the other inventory management system we decided to include some of common and key features that should be included in every inventory management system.

**Features:**

1. Login System
2. Add & Delete Inventory items
3. Search items

**Login system**

A user has to pass through a login system in order to get access. As application starts the login page appears. Talking about the system, it contains all the required functions which include adding, viewing, deleting and updating inventory items. Admin login is determined by the username and password that has all the authority to add, search and delete the stock of the organisation as per the requirement.

**Adding and Deleting inventory items:**

As one of the most important functionalities of the inventory management system these were also added. While adding inventory items, the user has to enter the product name, quantity, and its price. And when the details are entered the system shows the inventory record in a list view. And also the user easily delete any inventory items except the items only accessible to the admin. The user can search for an item as it contains a search function too.

**Search Items:**

This GUI based Simple Inventory system provides the simplest management of inventory items. In short, this projects mainly focus with a search function. There’s an external database connection file used to save the data permanently. As to do other tasks while keeping up with the requirements and making the required changes and adjustment the search function is a mandatory for most of the system as in this. This is a simple GUI Based system.

**Show items:**

This project also contains a module named show which displays all the products that are stored by the user/organization. This function simply put out the records in list view as mentioned above for the verification or any further tasks required.

**Future Scope**

This system can be used by the multiple peoples to get the counselling sessions online as it achieves efficiency and productivity in operations. Since it minimizes inventory costs and maximize sales &amp; profits and integrate entire business so it is an efficient way to develop many small businesses. Automation of manual tasks in the inventory can later integrate the different possibilities of AI and make the work process a lot more smother.

**Conclusion**

Inventory management system is an effective and easy to use tool. It is a quick response application that can store each details of every transaction. A proper development cycle must be followed for successful and timely completion of the project. The product must be thoroughly tested and according to the user requirement s all the functional and non-functional requirements of the system must be taken into consideration.

In nutshell, our project provides all the Details about the storage of products by any organization. It will be easier to find, add, search any product with the help of user-friendly interface.

**CODE**

**from tkinter import \***

**import tkinter.messagebox as tkMessageBox**

**import sqlite3**

**import tkinter.ttk as ttk**

**root = Tk()**

**root.title("HACKOVERFLOW INVENTORY SYSTEM")**

**width = 1300**

**height = 520**

**screen\_width = root.winfo\_screenwidth()**

**screen\_height = root.winfo\_screenheight()**

**x = (screen\_width/2) - (width/2)**

**y = (screen\_height/2) - (height/2)**

**root.geometry("%dx%d+%d+%d" % (width, height, x, y))**

**root.resizable(0, 0)**

**root.config(bg="azure4") #change the color**

**#========================================VARIABLES========================================**

**USERNAME = StringVar()**

**PASSWORD = StringVar()**

**PRODUCT\_NAME = StringVar()**

**PRODUCT\_PRICE = IntVar()**

**PRODUCT\_QTY = IntVar()**

**SEARCH = StringVar()**

**#========================================METHODS==========================================**

**def Database():**

**global conn, cursor**

**conn = sqlite3.connect("pythontut.db")**

**cursor = conn.cursor()**

**cursor.execute("CREATE TABLE IF NOT EXISTS `admin` (admin\_id INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL, username TEXT, password TEXT)")**

**cursor.execute("CREATE TABLE IF NOT EXISTS `product` (product\_id INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL, product\_name TEXT, product\_qty TEXT, product\_price TEXT)")**

**cursor.execute("SELECT \* FROM `admin` WHERE `username` = 'Krishanpal' AND `password` = 'Kp@123'")**

**if cursor.fetchone() is None:**

**cursor.execute("INSERT INTO `admin` (username, password) VALUES('Krishanpal', 'Kp@123')")**

**conn.commit()**

**def Exit():**

**result = tkMessageBox.askquestion('HACKOVERFLOW INVENTORY SYSTEM', 'Are you sure you want to exit?', icon="warning")**

**if result == 'yes':**

**root.destroy()**

**exit()**

**def Exit2():**

**result = tkMessageBox.askquestion('', 'Are you sure you want to exit?', icon="warning")**

**if result == 'yes':**

**Home.destroy()**

**exit()**

**def ShowLoginForm():**

**global loginform**

**loginform = Toplevel()**

**loginform.title("INVENTORY MANAGEMENT/ACCOUNT LOGIN")**

**width = 800**

**height = 400**

**screen\_width = root.winfo\_screenwidth()**

**screen\_height = root.winfo\_screenheight()**

**x = (screen\_width/2) - (width/2)**

**y = (screen\_height/2) - (height/2)**

**loginform.resizable(0, 0)**

**loginform.geometry("%dx%d+%d+%d" % (width, height, x, y))**

**LoginForm()**

**def LoginForm():**

**global lbl\_result**

**TopLoginForm = Frame(loginform, width=600, height=100, bd=1, relief=SOLID)**

**TopLoginForm.pack(side=TOP, pady=20)**

**lbl\_text = Label(TopLoginForm, text="Administrator Login", font=('arial', 18), width=600)**

**lbl\_text.pack(fill=X)**

**MidLoginForm = Frame(loginform, width=600)**

**MidLoginForm.pack(side=TOP, pady=50)**

**lbl\_username = Label(MidLoginForm, text="Username:", font=('arial', 25), bd=18)**

**lbl\_username.grid(row=0)**

**lbl\_password = Label(MidLoginForm, text="Password:", font=('arial', 25), bd=18)**

**lbl\_password.grid(row=1)**

**lbl\_result = Label(MidLoginForm, text="", font=('arial', 18))**

**lbl\_result.grid(row=3, columnspan=2)**

**username = Entry(MidLoginForm, textvariable=USERNAME, font=('arial', 25), width=15)**

**username.grid(row=0, column=1)**

**password = Entry(MidLoginForm, textvariable=PASSWORD, font=('arial', 25), width=15, show="\*")**

**password.grid(row=1, column=1)**

**btn\_login = Button(MidLoginForm, text="Login", font=('arial', 18), width=30, command=Login)**

**btn\_login.grid(row=2, columnspan=2, pady=20)**

**btn\_login.bind('<Return>', Login)**

**def Home():**

**global Home**

**Home = Tk()**

**Home.title("INVENTORY SYSTEM/HOME")**

**width = 1300**

**height = 520**

**screen\_width = Home.winfo\_screenwidth()**

**screen\_height = Home.winfo\_screenheight()**

**x = (screen\_width/2) - (width/2)**

**y = (screen\_height/2) - (height/2)**

**Home.geometry("%dx%d+%d+%d" % (width, height, x, y))**

**Home.resizable(0, 0)**

**Title = Frame(Home, bd=1, relief=SOLID)**

**Title.pack(pady=10)**

**lbl\_display = Label(Title, text="INVENTORY MANAGEMENT SYSTEM", font=('arial', 45))**

**lbl\_display.pack()**

**menubar = Menu(Home)**

**filemenu = Menu(menubar, tearoff=0)**

**filemenu2 = Menu(menubar, tearoff=0)**

**filemenu.add\_command(label="Logout", command=Logout)**

**filemenu.add\_command(label="Exit", command=Exit2)**

**filemenu2.add\_command(label="Add new", command=ShowAddNew)**

**filemenu2.add\_command(label="View", command=ShowView)**

**menubar.add\_cascade(label="Account", menu=filemenu)**

**menubar.add\_cascade(label="Inventory", menu=filemenu2)**

**Home.config(menu=menubar)**

**Home.config(bg="#6666ff")**

**def ShowAddNew():**

**global addnewform**

**addnewform = Toplevel()**

**addnewform.title("INVENTORY SYSTEM/ADD NEW")**

**width = 600**

**height = 500**

**screen\_width = Home.winfo\_screenwidth()**

**screen\_height = Home.winfo\_screenheight()**

**x = (screen\_width/2) - (width/2)**

**y = (screen\_height/2) - (height/2)**

**addnewform.geometry("%dx%d+%d+%d" % (width, height, x, y))**

**addnewform.resizable(0, 0)**

**AddNewForm()**

**def AddNewForm():**

**TopAddNew = Frame(addnewform, width=600, height=100, bd=1, relief=SOLID)**

**TopAddNew.pack(side=TOP, pady=20)**

**lbl\_text = Label(TopAddNew, text="Add New Product", font=('arial', 18), width=600)**

**lbl\_text.pack(fill=X)**

**MidAddNew = Frame(addnewform, width=600)**

**MidAddNew.pack(side=TOP, pady=50)**

**lbl\_productname = Label(MidAddNew, text="Product Name:", font=('arial', 25), bd=10)**

**lbl\_productname.grid(row=0, sticky=W)**

**lbl\_qty = Label(MidAddNew, text="Product Quantity:", font=('arial', 25), bd=10)**

**lbl\_qty.grid(row=1, sticky=W)**

**lbl\_price = Label(MidAddNew, text="Product Price:", font=('arial', 25), bd=10)**

**lbl\_price.grid(row=2, sticky=W)**

**productname = Entry(MidAddNew, textvariable=PRODUCT\_NAME, font=('arial', 25), width=15)**

**productname.grid(row=0, column=1)**

**productqty = Entry(MidAddNew, textvariable=PRODUCT\_QTY, font=('arial', 25), width=15)**

**productqty.grid(row=1, column=1)**

**productprice = Entry(MidAddNew, textvariable=PRODUCT\_PRICE, font=('arial', 25), width=15)**

**productprice.grid(row=2, column=1)**

**btn\_add = Button(MidAddNew, text="Save", font=('arial', 18), width=30, bg="#009ACD", command=AddNew)**

**btn\_add.grid(row=3, columnspan=2, pady=20)**

**def AddNew():**

**Database()**

**cursor.execute("INSERT INTO `product` (product\_name, product\_qty, product\_price) VALUES(?, ?, ?)", (str(PRODUCT\_NAME.get()), int(PRODUCT\_QTY.get()), int(PRODUCT\_PRICE.get())))**

**conn.commit()**

**PRODUCT\_NAME.set("")**

**PRODUCT\_PRICE.set("")**

**PRODUCT\_QTY.set("")**

**cursor.close()**

**conn.close()**

**def ViewForm():**

**global tree**

**TopViewForm = Frame(viewform, width=600, bd=1, relief=SOLID)**

**TopViewForm.pack(side=TOP, fill=X)**

**LeftViewForm = Frame(viewform, width=600)**

**LeftViewForm.pack(side=LEFT, fill=Y)**

**MidViewForm = Frame(viewform, width=600)**

**MidViewForm.pack(side=RIGHT)**

**lbl\_text = Label(TopViewForm, text="View Products", font=('arial', 18), width=600)**

**lbl\_text.pack(fill=X)**

**lbl\_txtsearch = Label(LeftViewForm, text="Search", font=('arial', 15))**

**lbl\_txtsearch.pack(side=TOP, anchor=W)**

**search = Entry(LeftViewForm, textvariable=SEARCH, font=('arial', 15), width=10)**

**search.pack(side=TOP, padx=10, fill=X)**

**btn\_search = Button(LeftViewForm, text="Search", command=Search)**

**btn\_search.pack(side=TOP, padx=10, pady=10, fill=X)**

**btn\_reset = Button(LeftViewForm, text="Reset", command=Reset)**

**btn\_reset.pack(side=TOP, padx=10, pady=10, fill=X)**

**btn\_delete = Button(LeftViewForm, text="Delete", command=Delete)**

**btn\_delete.pack(side=TOP, padx=10, pady=10, fill=X)**

**scrollbarx = Scrollbar(MidViewForm, orient=HORIZONTAL)**

**scrollbary = Scrollbar(MidViewForm, orient=VERTICAL)**

**tree = ttk.Treeview(MidViewForm, columns=("ProductID", "Product Name", "Product Qty", "Product Price"), selectmode="extended", height=100, yscrollcommand=scrollbary.set, xscrollcommand=scrollbarx.set)**

**scrollbary.config(command=tree.yview)**

**scrollbary.pack(side=RIGHT, fill=Y)**

**scrollbarx.config(command=tree.xview)**

**scrollbarx.pack(side=BOTTOM, fill=X)**

**tree.heading('ProductID', text="ProductID",anchor=W)**

**tree.heading('Product Name', text="Product Name",anchor=W)**

**tree.heading('Product Qty', text="Product Qty",anchor=W)**

**tree.heading('Product Price', text="Product Price",anchor=W)**

**tree.column('#0', stretch=NO, minwidth=0, width=0)**

**tree.column('#1', stretch=NO, minwidth=0, width=0)**

**tree.column('#2', stretch=NO, minwidth=0, width=200)**

**tree.column('#3', stretch=NO, minwidth=0, width=120)**

**tree.column('#4', stretch=NO, minwidth=0, width=120)**

**tree.pack()**

**DisplayData()**

**def DisplayData():**

**Database()**

**cursor.execute("SELECT \* FROM `product`")**

**fetch = cursor.fetchall()**

**for data in fetch:**

**tree.insert('', 'end', values=(data))**

**cursor.close()**

**conn.close()**

**def Search():**

**if SEARCH.get() != "":**

**tree.delete(\*tree.get\_children())**

**Database()**

**cursor.execute("SELECT \* FROM `product` WHERE `product\_name` LIKE ?", ('%'+str(SEARCH.get())+'%',))**

**fetch = cursor.fetchall()**

**for data in fetch:**

**tree.insert('', 'end', values=(data))**

**cursor.close()**

**conn.close()**

**def Reset():**

**tree.delete(\*tree.get\_children())**

**DisplayData()**

**SEARCH.set("")**

**def Delete():**

**if not tree.selection():**

**print("ERROR")**

**else:**

**result = tkMessageBox.askquestion('INVENTORY MANAGEMENT SYSTEM', 'Are you sure you want to delete this record?', icon="warning")**

**if result == 'yes':**

**curItem = tree.focus()**

**contents =(tree.item(curItem))**

**selecteditem = contents['values']**

**tree.delete(curItem)**

**Database()**

**cursor.execute("DELETE FROM `product` WHERE `product\_id` = %d" % selecteditem[0])**

**conn.commit()**

**cursor.close()**

**conn.close()**

**def ShowView():**

**global viewform**

**viewform = Toplevel()**

**viewform.title("INVENTORY SYSTEM/VIEW PRODUCT")**

**width = 600**

**height = 400**

**screen\_width = Home.winfo\_screenwidth()**

**screen\_height = Home.winfo\_screenheight()**

**x = (screen\_width/2) - (width/2)**

**y = (screen\_height/2) - (height/2)**

**viewform.geometry("%dx%d+%d+%d" % (width, height, x, y))**

**viewform.resizable(0, 0)**

**ViewForm()**

**def Logout():**

**result = tkMessageBox.askquestion('Inventory System', 'Are you sure you want to logout?', icon="warning")**

**if result == 'yes':**

**admin\_id = ""**

**root.deiconify()**

**Home.destroy()**

**def Login(event=None):**

**global admin\_id**

**Database()**

**if USERNAME.get == "" or PASSWORD.get() == "":**

**lbl\_result.config(text="Please complete the required field!", fg="red")**

**else:**

**cursor.execute("SELECT \* FROM `admin` WHERE `username` = ? AND `password` = ?", (USERNAME.get(), PASSWORD.get()))**

**if cursor.fetchone() is not None:**

**cursor.execute("SELECT \* FROM `admin` WHERE `username` = ? AND `password` = ?", (USERNAME.get(), PASSWORD.get()))**

**data = cursor.fetchone()**

**admin\_id = data[0]**

**USERNAME.set("")**

**PASSWORD.set("")**

**lbl\_result.config(text="")**

**ShowHome()**

**else:**

**lbl\_result.config(text="Invalid username or password", fg="red")**

**USERNAME.set("")**

**PASSWORD.set("")**

**cursor.close()**

**conn.close()**

**def ShowHome():**

**root.withdraw()**

**Home()**

**loginform.destroy()**

**#========================================MENUBAR WIDGETS==================================**

**menubar = Menu(root)**

**filemenu = Menu(menubar, tearoff=0)**

**filemenu.add\_command(label="Account", command=ShowLoginForm)**

**filemenu.add\_command(label="Exit", command=Exit)**

**menubar.add\_cascade(label="File", menu=filemenu)**

**root.config(menu=menubar)**

**#========================================FRAME============================================**

**Title = Frame(root, bd=1, relief=SOLID)**

**Title.pack(pady=10)**

**#========================================LABEL WIDGET=====================================**

**lbl\_display = Label(Title, text="INVENTORY MANAGEMENT SYSTEM", font=('arial', 45))**

**lbl\_display.pack()**

**#========================================INITIALIZATION===================================**

**if \_\_name\_\_ == '\_\_main\_\_':**

**root.mainloop()**