Objectives:

* TKINTER Pharmacy Application.

**There is one print screen worth 100%**

**Project**

The pharmacy application will have the ability to display, enter, update, and delete certain medicines.

Creating the table. (Ensure that the database is already created).

A close-up of a white background

Description automatically generated

Creating the TKINTER Pharmacy Application.

A screen shot of a computer program

Description automatically generatedA computer screen shot of a program

Description automatically generatedA computer screen shot of a program

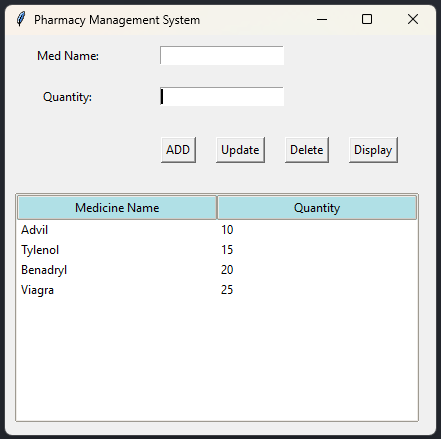
Description automatically generated

A computer screen shot of a program code

Description automatically generatedA black screen with green and white text

Description automatically generated

**Print screen the running application below here.**

****

**Copy and paste your code below here.**

Code:

import tkinter as tk

from tkinter import messagebox

from tkinter import ttk

import sqlite3 as sql

class PharmacyManagementSystems:

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

self.window = tk.Tk()

self.window.title("Pharmacy Management System")

self.medicines = [] #Capture Medicine Input

self.name\_label = tk.Label(self.window, text="Med Name:")

self.name\_label.grid(row=0, column=0, padx=10, pady=10)

self.name\_entry = tk.Entry(self.window)

self.name\_entry.grid(row=0, column=1, padx=10, pady=10)

self.qty\_label = tk.Label(self.window, text="Quantity:")

self.qty\_label.grid(row=1, column=0, padx=10, pady=10)

self.qty\_entry = tk.Entry(self.window)

self.qty\_entry.grid(row=1, column=1, padx=10, pady=10)

frm = tk.Frame(self.window)

frm.grid(row=2, column=1, columnspan=4, pady=10)

self.add\_button = tk.Button(frm, text="ADD", command=self.add\_medicine)

self.add\_button.grid(row=0, column=0, padx=10, pady=10)

self.update\_button = tk.Button(frm, text="Update", command=self.update\_medicine)

self.update\_button.grid(row=0, column=1, padx=10, pady=10)

self.delete\_button = tk.Button(frm, text="Delete", command=self.delete\_medicine)

self.delete\_button.grid(row=0, column=2, padx=10, pady=10)

self.display\_button = tk.Button(frm, text="Display", command=self.display\_medicines)

self.display\_button.grid(row=0, column=3, padx=10, pady=10)

cols = ["Name", "Qty"]

self.treeview = ttk.Treeview(self.window, column=cols, show='headings', height=10)

self.treeview.grid(row=3, column=0, padx=10, pady=10, columnspan=5)

style = ttk.Style()

style.theme\_use('clam')

style.configure("Treeview", background="white", fieldbackground="white", foreground="black")

style.configure('Treeview.Heading', background="PowderBlue")

self.treeview.heading("Name", text="Medicine Name")

self.treeview.heading("Qty", text="Quantity")

self.conn = sql.connect("C:\\Users\\17147\\Desktop\\SQLLite\\PythonClassDB.db")

self.cursor = self.conn.cursor()

def add\_medicine(self):

name = self.name\_entry.get()

quantity = self.qty\_entry.get()

if name and quantity:

self.medicines.append((name, quantity))

messagebox.showinfo(title="Success", message="Inserted successfully.")

self.name\_entry.delete(0, tk.END)

self.qty\_entry.delete(0, tk.END)

self.save\_medicine\_to\_database(name, quantity)

self.display\_medicines()

else:

messagebox.showerror(title="Error", message="Please both fields.")

def update\_medicine(self):

name = self.name\_entry.get()

quantity = self.qty\_entry.get()

if name and quantity:

#self.medicines.append((name, quantity))

# messagebox.showinfo(title="Success", message="Updated successfully.")

# self.name\_entry.delete(0, tk.END)

# self.qty\_entry.delete(0, tk.END)

self.update\_medicine\_to\_database(name, quantity)

self.display\_medicines()

else:

messagebox.showerror(title="Error", message="Please enter both fields.")

def delete\_medicine(self):

name = self.name\_entry.get()

if name:

#self.medicines.remove(name)

messagebox.showinfo(title="Success", message="Deleted successfully.")

self.name\_entry.delete(0, tk.END)

self.qty\_entry.delete(0, tk.END)

self.delete\_medicine\_to\_database(name)

self.display\_medicines()

else:

messagebox.showerror(title="Error", message="Please enter name field.")

def save\_medicine\_to\_database(self, name, quantity):

try:

self.cursor.execute("Insert into medicines (name, quantity) values (?,?)", (name, quantity))

self.conn.commit()

except sql.Error as e:

messagebox.showerror(title="Error", message=f"Database Error: {e}")

def update\_medicine\_to\_database(self, name, quantity):

try:

self.cursor.execute("UPDATE medicines SET quantity=? WHERE name = ?)", (quantity, name))

self.conn.commit()

except sql.Error as e:

messagebox.showerror(title="Error", message=f"Database Error: {e}")

def delete\_medicine\_to\_database(self, name):

try:

self.cursor.execute("DELETE medicines WHERE name =?", (name))

self.conn.commit()

except sql.Error as e:

messagebox.showerror(title="Error", message=f"Database Error: {e}")

def display\_medicines(self):

try:

self.treeview.delete(\*self.treeview.get\_children())

self.cursor.execute("SELECT name, quantity FROM medicines")

medicines = self.cursor.fetchall()

for medicine in medicines:

self.treeview.insert("", tk.END, values=medicine)

except sql.Error as e:

messagebox.showerror(title="Error", message=f"Database Error: {e}")

def \_\_del\_\_(self):

self.cursor.close()

self.conn.close()

def run(self):

self.window.geometry("600x400")

self.window.mainloop()

phar\_sys = PharmacyManagementSystems()

phar\_sys.run()

**Submit this document to the Module 6 Class Exercise.**