EX.NO:	**HOSPITAL INVENTORY MANAGEMENT USING TKINTER**
DATE: / /2025	

AIM

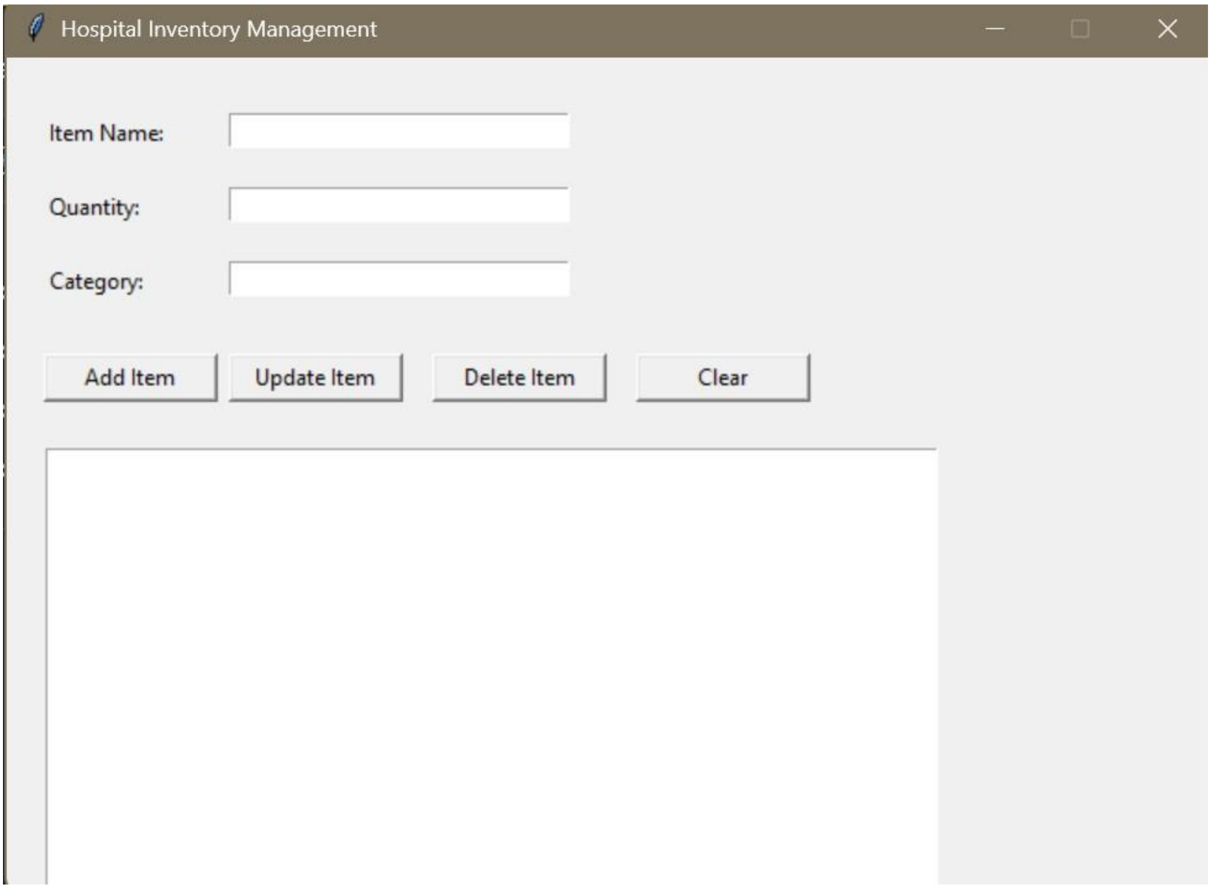
To Develop a Hospital Inventory Management using tkinter in Python

SOURCE CODE:

```
import tkinter as tk
from tkinter import messagebox
import sqlite3
def connect_db():
    conn = sqlite3.connect("hospital_inventory.db")
    cur = conn.cursor()
    cur.execute("""
        CREATE TABLE IF NOT EXISTS inventory (
            id INTEGER PRIMARY KEY AUTOINCREMENT,
            item_name TEXT NOT NULL,
            quantity INTEGER NOT NULL,
            category TEXT NOT NULL
        )
    """)
    conn.commit()
    conn.close()
def add_item():
    name = entry_name.get()
    qty = entry_qty.get()
    category = entry_category.get()
    if name == "" or qty == "" or category == "":
        messagebox.showerror("Error", "All fields are required!")
        return
    conn = sqlite3.connect("hospital_inventory.db")
    cur = conn.cursor()
    cur.execute("INSERT INTO inventory (item_name, quantity, category) VALUES (?, ?, ?)",
                (name, qty, category))
    conn.commit()
    conn.close()
    view_items()
    clear_fields()
    messagebox.showinfo("Success", "Item added successfully!")
```

```
def view_items():
    listbox_items.delete(0, tk.END)
    conn = sqlite3.connect("hospital_inventory.db")
    cur = conn.cursor()
    cur.execute("SELECT * FROM inventory")
    rows = cur.fetchall()
    conn.close()
    for row in rows:
        listbox_items.insert(tk.END, row)
def delete_item():
    try:
        selected = listbox_items.get(listbox_items.curselection())
        item_id = selected[0]
    except:
        messagebox.showerror("Error", "Select an item to delete!")
        return
    conn = sqlite3.connect("hospital_inventory.db")
    cur = conn.cursor()
    cur.execute("DELETE FROM inventory WHERE id=?", (item_id,))
    conn.commit()
    conn.close()
    view_items()
    clear_fields()
    messagebox.showinfo("Success", "Item deleted successfully!")
def update_item():
    try:
        selected = listbox_items.get(listbox_items.curselection())
        item_id = selected[0]
    except:
        messagebox.showerror("Error", "Select an item to update!")
        return
    name = entry_name.get()
    qty = entry_qty.get()
    category = entry_category.get()
    if name == "" or qty == "" or category == "":
        messagebox.showerror("Error", "All fields are required!")
        return
    conn = sqlite3.connect("hospital_inventory.db")
    cur = conn.cursor()
    cur.execute("""
        UPDATE inventory
        SET item_name=?, quantity=?, category=?
        WHERE id=?
    """, (name, qty, category, item_id))
    conn.commit()
    conn.close()
    view_items()
    clear_fields()
    messagebox.showinfo("Success", "Item updated!")
def fill_fields(event):
    try:
        selected = listbox_items.get(listbox_items.curselection())
        entry_name.delete(0, tk.END)
        entry_qty.delete(0, tk.END)
        entry_category.delete(0, tk.END)
        entry_name.insert(tk.END, selected[1])
        entry_qty.insert(tk.END, selected[2])
        entry_category.insert(tk.END, selected[3])
    except:
        pass
def clear_fields():
    entry_name.delete(0, tk.END)
    entry_qty.delete(0, tk.END)
    entry_category.delete(0, tk.END)
root = tk.Tk()
root.title("Hospital Inventory Management")
root.geometry("650x450")
root.resizable(False, False)
tk.Label(root, text="Item Name:").place(x=20, y=30)
tk.Label(root, text="Quantity:").place(x=20, y=70)
tk.Label(root, text="Category:").place(x=20, y=110)
entry_name = tk.Entry(root, width=30)
entry_name.place(x=120, y=30)
entry_qty = tk.Entry(root, width=30)
entry_qty.place(x=120, y=70)
entry_category = tk.Entry(root, width=30)
entry_category.place(x=120, y=110)
tk.Button(root, text="Add Item", width=12, command=add_item).place(x=20, y=160)
tk.Button(root, text="Update Item", width=12, command=update_item).place(x=120, y=160)
tk.Button(root, text="Delete Item", width=12, command=delete_item).place(x=230, y=160)
tk.Button(root, text="Clear", width=12, command=clear_fields).place(x=340, y=160)
listbox_items = tk.Listbox(root, width=80, height=15)
listbox_items.place(x=20, y=210)
listbox_items.bind("<<ListboxSelect>>", fill_fields)
connect_db()
view_items()
root.mainloop()
```

OUTPUT:



RESULT:

The program has been Successfully executed.