EX.NO:	**HOSPITAL DISCHARGE APPLICATION USING STACK**
DATE: / /2025	

AIM

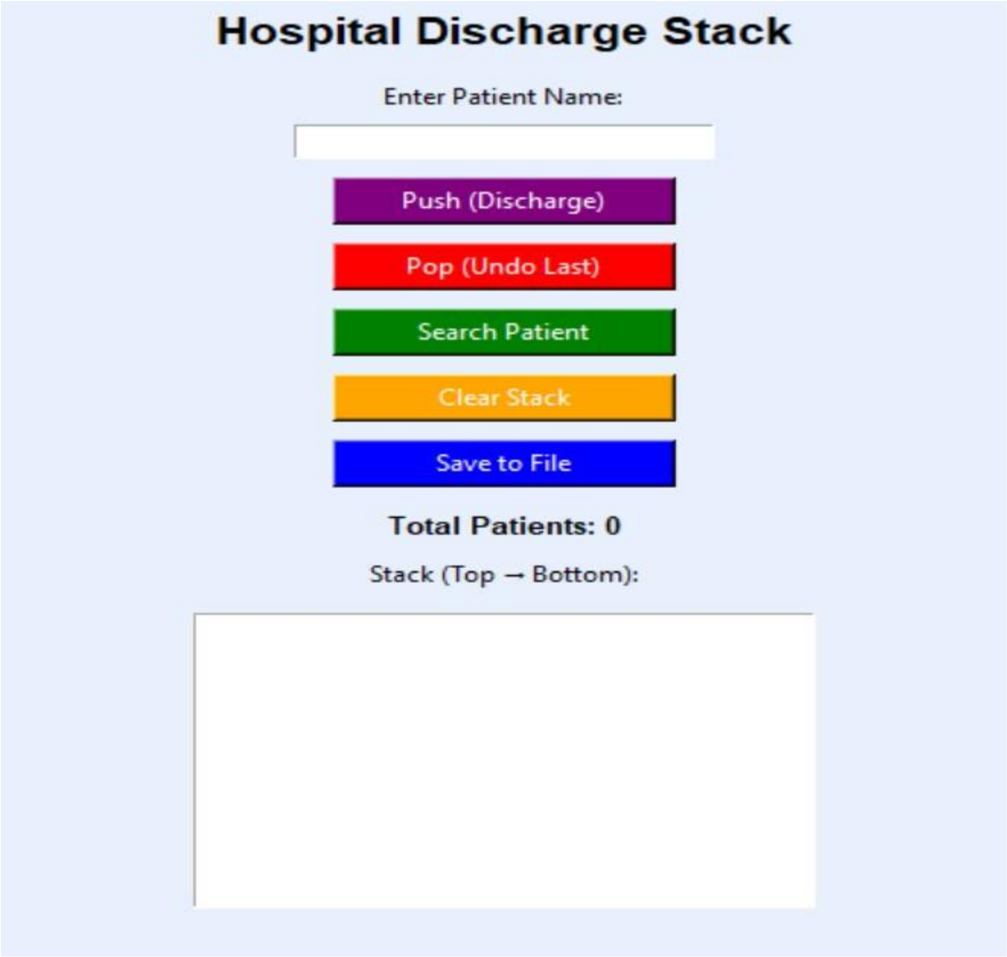
To Develop hospital discharge application using stack in python.

SOURCE CODE:

```
import tkinter as tk
from tkinter import messagebox, simpledialog
from datetime import datetime
discharge_stack =
def push_patient(event=None):  # Added event for Enter key
    name = entry_name.get().strip()
    if name == "":
        messagebox.showwarning("Warning", "Enter patient name")
        return
    timestamp = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
    discharge_stack.append(f"{name} (at {timestamp})")
    messagebox.showinfo("Discharged", f"Patient discharged:\n{name}")
    entry_name.delete(0, tk.END)
    show_stack()
def pop_patient():
    if not discharge_stack:
        messagebox.showwarning("Empty Stack", "No patients discharged")
        return
    patient = discharge_stack.pop()
    messagebox.showinfo("Undo", f"Removed from stack:\n{patient}")
    show_stack()
def show_stack():
    listbox.delete(0, tk.END)
    for p in reversed(discharge_stack):
        listbox.insert(tk.END, p)
    lbl_count.config(text=f"Total Patients: {len(discharge_stack)}")
def clear_stack():
    if not discharge_stack:
        messagebox.showwarning("Warning", "Stack already empty")
        return

    if messagebox.askyesno("Confirm", "Clear entire stack?"):
        discharge_stack.clear()
        show_stack()
def search_patient():
    query = simpledialog.askstring("Search", "Enter patient name to search:")
    if not query:
        return
    matches = [p for p in discharge_stack if query.lower() in p.lower()]
    if matches:
        messagebox.showinfo("Found", "\n".join(matches))
    else:
        messagebox.showinfo("Not Found", "No patient found")
def save_to_file():
    if not discharge_stack:
        messagebox.showwarning("Warning", "No data to save")
        return
    with open("discharge_history.txt", "w") as file:
        for p in discharge_stack:
            file.write(p + "\n") messagebox.showinfo("Saved", "History saved as
discharge_history.txt")
root.title("Hospital Patient Stack System")
root.geometry("450x450")
root.config(bg="#E8F0FE")
tk.Label(root, text="Hospital Discharge Stack",
         font=("Arial", 16, "bold"), bg="#E8F0FE").pack(pady=10)
tk.Label(root, text="Enter Patient Name:", bg="#E8F0FE").pack()
entry_name = tk.Entry(root, width=30)
entry_name.pack(pady=5)
entry_name.bind("<Return>", push_patient)  # Press Enter to push
tk.Button(root, text="Push (Discharge)", width=20, bg="purple", fg="white",
          command=push_patient).pack(pady=5)
tk.Button(root, text="Pop (Undo Last)", width=20, bg="red", fg="white",
          command=pop_patient).pack(pady=5
tk.Button(root, text="Search Patient", width=20, bg="green", fg="white",
          command=search_patient).pack(pady=5)
tk.Button(root, text="Clear Stack", width=20, bg="orange", fg="white",
          command=clear_stack).pack(pady=5)
tk.Button(root, text="Save to File", width=20, bg="blue", fg="white",
        command=save_to_file).pack(pady=5)
lbl_count = tk.Label(root, text="Total Patients: 0",bg="#E8F0FE", font=("Arial"))
lbl_count.pack(pady=5)
tk.Label(root, text="Stack (Top → Bottom):", bg="#E8F0FE").pack()
listbox.pack(pady=10)
root.mainloop()
```

OUTPUT:



Total Patients: 0

RESULT:

The program has been Successfully Executed.