**TO-DO Gui**

**Sarvesh Shashikant Sawant**

Department of Electronics Engineering

Shah and Anchor Kutchhi Engineering College

Mumbai: 400088, India

sarvesh.sawant\_19@sakec.ac.in

**Abstract: -** *This paper illustrates configuration of “TO-DO” Gui that helps an individual to keep a track on his/her daily activities. Every individual has a lot of choices to help them in keeping track of daily obligations. A simple list on paper of things “To Do” is enough for some people. Others prefer to use programs. People have preferences about where they keep track of tasks - with PC utilities, websites, on phone apps. In order to keep a track of our daily activities or setting up our tasks, the “TO-DO” Gui can be very much useful. This paper discloses on how the “TO-DO” Gui is configured and programmed.*

**Introduction: -** Human beings have a tendency to forget a lot of things on daily basis, so to keep a track of it some write it on a piece of paper, some writes it in a book and in the modern world where we have smartphones lot of peoples save it on their TODO apps. So to solve this problem, a “TO-DO” Gui has been developed. In this “TO-DO” Gui any individual can list out his/her tasks that he/she wants to perform. It provides the feature of saving and loading his/her data which will help them to continue their tasks in coming future without worrying about loss of any task related data. After completion of the task, the provided data can be deleted also.

**Algorithm: -**

Step 1: START

Step 2: Import Tkinter, tkMessageBox and Pickle module.

Step 3: Create root window with title as “TO-DO List”.

Step 4: Define Functions in order to perform Add tasks, Delete tasks, Save tasks, Load tasks, Clear all and Clear entry.

Step 5: Create GUI.

Step 6: In GUI create a Frame.

Step 7: Create a List box for displaying our tasks.

Step 8: Create a vertical scrollbar adjacent to the List box.

Step 9: Create an entry field to enter our tasks.

Step 10: Create buttons for each functions

Step 11: Give Commands to all the buttons.

Step 12: END

**CODE with comments: -**

#import libraries

import tkinter

import tkinter.messagebox

import pickle

#create root window

root = tkinter.Tk()

#give title

root.title("TO-DO List ")

#give background color

root.config(bg="black")

#Create functions

def add\_task():

task= entry\_task.get()

if task != "":

listbox\_tasks.insert(tkinter.END, task)

entry\_task.delete(0, tkinter.END)

else:

tkinter.messagebox.showwarning(title="Warning!", message=" You must enter a task.")

def delete\_task():

try:

task\_index = listbox\_tasks.curselection()[0]

listbox\_tasks.delete(task\_index)

tkinter.messagebox.showwarning(title="Delete", message=" Your selected task is DELETED.")

except:

tkinter.messagebox.showwarning(title="Warning!", message="You must select a task.")

def clear\_all():

listbox\_tasks.delete(0, tkinter.END)

tkinter.messagebox.showwarning(title="Clear all", message="All tasks are cleared.")

def load\_tasks():

try:

tasks = pickle.load(open("tasks.dat", "rb"))

listbox\_tasks.delete(0, tkinter.END)

for task in tasks:

listbox\_tasks.insert(tkinter.END, task)

except:

tkinter.messagebox.showwarning(title="Warning!", message="Cannot find tasks.dat.")

def save\_tasks():

tasks = listbox\_tasks.get(0, listbox\_tasks.size())

pickle.dump(tasks, open("tasks.dat", "wb"))

def clear\_entry(event,entry\_task):

entry\_task.delete(0,tkinter.END)

#Creation of GUI

#Create Frame

frame\_tasks = tkinter.Frame(root)

frame\_tasks.pack()

#Create a Listbox for displaying our tasks

listbox\_tasks = tkinter.Listbox(frame\_tasks, height=10, width=50, fg="black", bg="grey", bd=4)

listbox\_tasks.pack(side=tkinter.LEFT)

#Create scroll bar

scrollbar\_tasks = tkinter.Scrollbar(frame\_tasks)

scrollbar\_tasks.pack(side=tkinter.RIGHT, fill=tkinter.Y)

listbox\_tasks.config(yscrollcommand=scrollbar\_tasks.set)

scrollbar\_tasks.config(command=listbox\_tasks.yview)

#Create Entry field to enter our tasks

entry\_task = tkinter.Entry(root, width=50)

entry\_task.insert(0,"Enter Task Here...")

entry\_task.bind("<Button-1>", lambda event: clear\_entry(event, entry\_task) )

entry\_task.pack()

#Create Buttons and give commands

button\_add\_task = tkinter.Button(root, text="Add task", width=48, bg="grey",fg="black",bd=4, command=add\_task)

button\_add\_task.pack()

button\_save\_tasks = tkinter.Button(root, text="Save tasks", width=48, bg="grey",fg="black",bd=4, command=save\_tasks)

button\_save\_tasks.pack()

button\_load\_tasks = tkinter.Button(root, text="Load tasks", width=48, bg="grey",fg="black", bd=4, command=load\_tasks)

button\_load\_tasks.pack()

button\_delete\_task = tkinter.Button(root, text="Delete task", width=48, bg="grey",fg="black",bd=4, command=delete\_task)

button\_delete\_task.pack()

button\_Clear\_all\_tasks = tkinter.Button(root, text="Clear all tasks", width=48, bg="grey",fg="black",bd=4, command=clear\_all)

button\_Clear\_all\_tasks.pack()

button\_Exit = tkinter.Button(root, text="Exit", width=48, bg="grey",fg="red",bd=4, command=root.destroy)

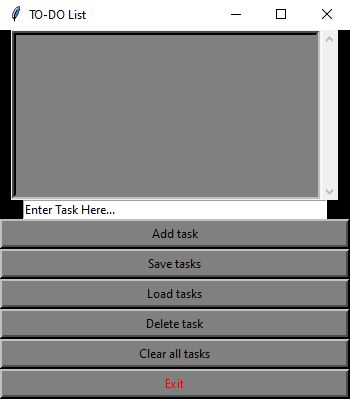
button\_Exit.pack()

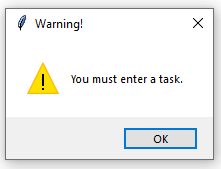
root.mainloop()

**Result and Analysis: -**

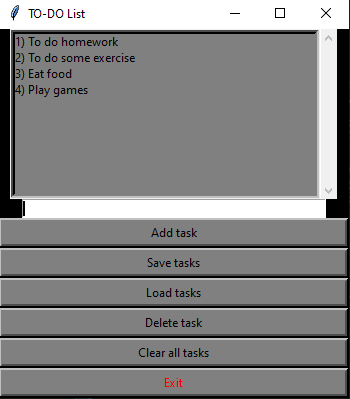
**(Output of Program as Result with Analysis)**

1. Popup message if add task button is pressed without entering tasks

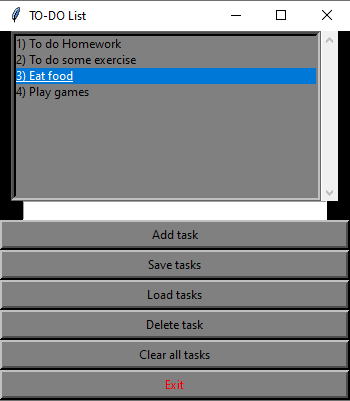
****

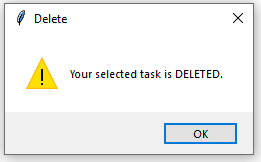
****

1. Adding particular tasks

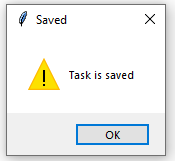


1. Deleting specific task

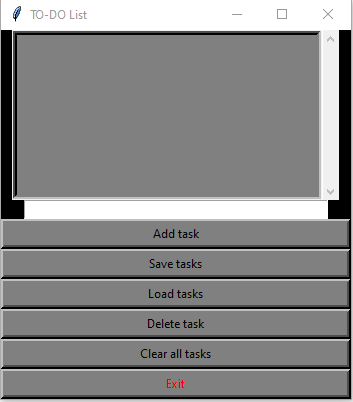


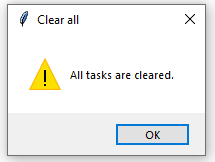


1. Message box after the task is saved



1. Clearing all tasks





The tasks entered in the entry box is displayed in the list box after clicking the ‘add task’ button. The ‘save task’ button save the task and by clicking the ‘load task’ button, the saved task is displayed on the list box. The entered list of task can be deleted individually by selecting the specific task and clicking on the ‘delete task ’ button. The ‘clear all’ button clears the entire list box and the ‘exit’ button exits the window.

**Conclusion: -** The “TO-DO List” GUI is solely based on its ability to perform the function of listing out various tasks that an individual want to complete on daily basis. Implementation of “TO-DO List” GUI would eliminate the problems faced by each and every individual of forgetting their daily activities. This paper elaborates the idea and making of “TO-DO List” GUI and also explains about the successful working of the program by performing all the required functions.

**References: -**

1. <https://www.geeksforgeeks.org/python-gui-tkinter/>
2. <https://docs.python.org/3/library/pickle.html#:~:text=%E2%80%9CPickling%E2%80%9D%20is%20the%20process%20whereby,back%20into%20an%20object%20hierarchy>.
3. <https://www.geeksforgeeks.org/python-binding-function-in-tkinter/#:~:text=The%20binding%20function%20is%20used,mouse%20movement%20with%20tkinter%20Frame>.
4. <https://www.youtube.com/watch?v=8qUJ9a_3zSQ>