**(Name – Rutuja Salunkhe)**

**Task 1- To do list**

**A To-Do List application is a useful project that helps users manage**

**and organize their tasks efficiently. This project aims to create a**

**command-line or GUI-based application using Python, allowing**

**users to create, update, and track their to-do lists**

Code-

import tkinter as tk

from tkinter import messagebox

class TodoApp:

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

self.root = root

self.root.title("Colorful To-Do List")

self.root.geometry("400x400")

# Set the background color

self.root.config(bg="#f7f5f2")

# Title Label

self.title\_label = tk.Label(self.root, text="My To-Do List", bg="#f7f5f2", fg="#ff6347", font=("Helvetica", 18, "bold"))

self.title\_label.pack(pady=10)

# Entry Frame

self.entry\_frame = tk.Frame(self.root, bg="#f7f5f2")

self.entry\_frame.pack(pady=10)

self.task\_entry = tk.Entry(self.entry\_frame, width=30, font=("Helvetica", 12))

self.task\_entry.pack(side=tk.LEFT, padx=10)

self.add\_button = tk.Button(self.entry\_frame, text="Add Task", command=self.add\_task, bg="#4682b4", fg="#ffffff", font=("Helvetica", 12))

self.add\_button.pack(side=tk.LEFT)

# Task List Frame

self.list\_frame = tk.Frame(self.root, bg="#f7f5f2")

self.list\_frame.pack(pady=10)

self.task\_listbox = tk.Listbox(self.list\_frame, width=50, height=10, font=("Helvetica", 12), bg="#fffacd", fg="#4682b4", selectbackground="#ff6347", selectforeground="#ffffff")

self.task\_listbox.pack(side=tk.LEFT, fill=tk.BOTH)

self.scrollbar = tk.Scrollbar(self.list\_frame)

self.scrollbar.pack(side=tk.RIGHT, fill=tk.BOTH)

self.task\_listbox.config(yscrollcommand=self.scrollbar.set)

self.scrollbar.config(command=self.task\_listbox.yview)

# Delete Button

self.delete\_button = tk.Button(self.root, text="Delete Selected", command=self.delete\_task, bg="#ff6347", fg="#ffffff", font=("Helvetica", 12))

self.delete\_button.pack(pady=20)

def add\_task(self):

task = self.task\_entry.get()

if task != "":

self.task\_listbox.insert(tk.END, task)

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

else:

messagebox.showwarning("Warning", "You must enter a task.")

def delete\_task(self):

try:

selected\_task\_index = self.task\_listbox.curselection()[0]

self.task\_listbox.delete(selected\_task\_index)

except:

messagebox.showwarning("Warning", "You must select a task to delete.")

if \_\_name\_\_ == "\_\_main\_\_":

root = tk.Tk()

app = TodoApp(root)

root.mainloop()