import pickle

from tkinter import \*

from tkinter.font import Font

from tkinter import filedialog

root = Tk()

root.title('To-Do-List-App')

root.geometry("500x500")

# Define our font

my\_font = Font(

family="Brush Script MT",

size=30,

weight="bold")

# Create frame

my\_frame = Frame(root)

my\_frame.pack(pady=10)

# Create listbox

my\_list = Listbox(my\_frame,

font=my\_font,

width=25,

height=5,

bg="#EBEDEF",

bd=0,

fg="#464646", borderwidth=5

, highlightthickness=0,

selectbackground="#a6a6a6",

activestyle="none"

) # systemButtonFace

my\_list.pack()

# Create dummy list

stuff = ["Wake up at 6 am","Go to College", "Buy notebook", "Came to home", "Complete the project", "Complete task2",

"Take medicines"]

# Add dummy list to the list box

for item in stuff:

my\_list.insert(END, item)

# Add scroll bar

my\_scrollbar = Scrollbar(my\_frame)

my\_scrollbar.pack(side=RIGHT, fill=BOTH)

# add scrollbar

my\_list.config(yscrollcommand=my\_scrollbar.set)

my\_scrollbar.config(command=my\_list.yview)

# create an entry box to add items to the list

my\_entry = Entry(root, font=("Helvetica", 24), width=26, borderwidth=5)

my\_entry.pack(pady=20)

# create button frame

button\_frame = Frame(root)

button\_frame.pack(pady=20)

# Functions

def delete\_item():

my\_list.delete(ANCHOR)

def add\_item():

my\_list.insert(END, my\_entry.get())

my\_entry.delete(0, END)

def cross\_off\_item():

# cross off item

my\_list.itemconfig(

my\_list.curselection(),

fg="#dedede")

# get rid of selection bar

my\_list.select\_clear(0, END)

def uncross\_item():

# cross off item

my\_list.itemconfig(

my\_list.curselection(),

fg="#464646")

# get rid of selection bar

my\_list.select\_clear(0, END)

def delete\_crossed():

count = 0

while count < my\_list.size():

if my\_list.itemcget(count, "fg") == "#dedede":

my\_list.delete(my\_list.index(count))

else:

count += 1

def save\_list():

file\_name = filedialog.asksaveasfilename(

initialdir="D:\Python\_Projects\info\_aidtech\_task\data"

, title="Save File", filetypes=(("Dat Files", "\*.dat")

, ("All Files", "\*.\*")))

if file\_name:

if file\_name.endswith(".dat"):

pass

else:

file\_name = f'{file\_name}.dat'

# Delete crossed off items before saving

count = 0

while count < my\_list.size():

if my\_list.itemcget(count, "fg") == "#dedede":

my\_list.delete(my\_list.index(count))

else:

count += 1

# Grab all the stuff from the list

my\_list.get(0, END)

# Open the file

output\_file = open(file\_name, 'wb')

# Actually add the stuff to the file

pickle.dump(stuff, output\_file)

def open\_list():

file\_name = filedialog.askopenfilename(

initialdir="D:\Python\_Projects\info\_aidtech\_task\data"

, title="Save File", filetypes=(("Dat Files", "\*.dat")

, ("All Files", "\*.\*")))

if file\_name:

# Delete currently open list

my\_list.delete(0, END)

input\_file = open(file\_name, 'rb')

# Load the data the data from the file

stuff = pickle.load(input\_file)

# Output the stuff to the screen

for item in stuff:

my\_list.insert(END, item)

def clear\_list():

my\_list.delete(0, END)

# Create Menu

my\_menu = Menu(root)

root.config(menu=my\_menu, bg="#808B96")

# Add items to the menu

file\_menu = Menu(my\_menu, tearoff=False)

my\_menu.add\_cascade(label="File", menu=file\_menu)

# Add drop down items

file\_menu.add\_command(label="Save List", command=save\_list)

file\_menu.add\_command(label="Open List", command=open\_list)

file\_menu.add\_separator()

file\_menu.add\_command(label="Clear List", command=clear\_list)

# Add some buttons

delete\_button = Button(button\_frame, text="Delete Item", command=delete\_item)

add\_button = Button(button\_frame, text="Add Item", command=add\_item)

cross\_off\_button = Button(button\_frame, text="Cross off Item", command=cross\_off\_item)

uncross\_button = Button(button\_frame, text="Uncross Item", command=uncross\_item)

delete\_crossed\_button = Button(button\_frame, text="Delete Crossed", command=delete\_crossed)

delete\_button.grid(row=0, column=0)

add\_button.grid(row=0, column=1, padx=20)

cross\_off\_button.grid(row=0, column=2)

uncross\_button.grid(row=0, column=3, padx=20)

delete\_crossed\_button.grid(row=0, column=4)

root.mainloop()