Ex:2 - CLI , GUI AND VUI :

1) CLI :

```
tasks = []
def add_task(task):
 """Adds a task to the list."""
 tasks.append(task)
 print(f"Task '{task}' added.")

def view_tasks():
 """Displays the tasks."""
 if tasks:
     print("\nYour tasks:")
     for idx, task in enumerate(tasks, 1):
         print(f"{idx}. {task}")
 else:
     print("\nNo tasks to show.")

def remove_task(task_number):
 """Removes a task by its number."""
 if 0 < task_number <= len(tasks):
     removed_task = tasks.pop(task_number - 1)
     print(f"Task '{removed_task}' removed.")
 else:
     print("Invalid task number.")

def main():
 """Main function to handle user input."""
 while True:
     print("\nOptions: ")
     print("1. Add Task")
     print("2. View Tasks")
     print("3. Remove Task")
     print("4. Exit")
     choice = input("Enter your choice: ")
     if choice == '1':
         task = input("Enter task: ")
         add_task(task)
     elif choice == '2':
         view_tasks()
     elif choice == '3':
         try:
             task_number = int(input("Enter task number to remove: "))
             remove_task(task_number)
         except ValueError:
             print("Invalid input. Please enter a valid number.")
     elif choice == '4':
         print("Exiting...")
         break
```

```
Options:
1. Add Task
2. View Tasks
3. Remove Task
4. Exit
Enter your choice: 1
Enter task: Write Homework
Task 'Write Homework' added.

Options:
1. Add Task
2. View Tasks
3. Remove Task
4. Exit
Enter your choice: 2

Your tasks:
1. Write Homework

Options:
1. Add Task
2. View Tasks
3. Remove Task
4. Exit
Enter your choice: 3
Enter task number to remove: 1
Task 'Write Homework' removed.

Options:
1. Add Task
2. View Tasks
3. Remove Task
4. Exit
Enter your choice: 4
Exiting...
```

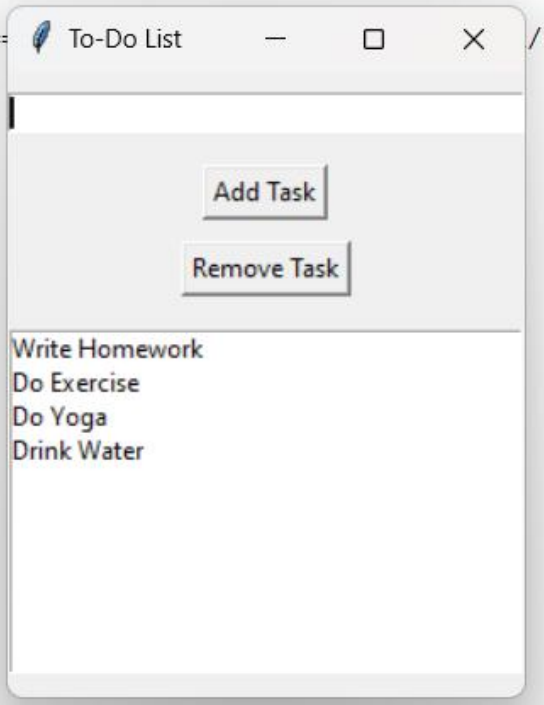
2) GUI :

UID 3A.py - C:/Users/A.Velan/Downloads/UID 3A.py (3.12.0)

File Edit Format Run Options Window Help

```
import tkinter as tk
from tkinter import messagebox

tasks = []
def add_task():
 """Adds a task to the list."""
 task = task_entry.get()

 if task:
 tasks.append(task)
 task_entry.delete(0, tk.END)
 update_task_list()
 else:
 messagebox.showwarning("Warning", "Task cannot be empty")
def update_task_list():
 """Updates the task list display."""
 task_list.delete(0, tk.END)
 for task in tasks:
 task_list.insert(tk.END, task)
def remove_task():
 """Removes the selected task from the list."""
 selected_task_index = task_list.curselection()

 if selected_task_index:
 task_list.delete(selected_task_index)
 tasks.pop(selected_task_index[0])
 else:
 messagebox.showwarning("Warning", "Please select a task to remove")

app = tk.Tk()
app.title("To-Do List")

task_entry = tk.Entry(app, width=40)
task_entry.pack(pady=10)

add_button = tk.Button(app, text="Add Task", command=add_task)
add_button.pack(pady=5)

remove_button = tk.Button(app, text="Remove Task", command=remove_task)
remove_button.pack(pady=5)

task_list = tk.Listbox(app, width=40, height=10)
task_list.pack(pady=10)

app.mainloop()
```

```
Python 3.12.0 (tags/v3.12.0:0fb18b0, Oct  2 2023, 13:03:39) [MSC
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more infor

= RESTART: C:/Users/A.Velan/Downloads/UID 3A.py

==========
```




```
/Downloads/UID 3A.py =
```

3) VUI :

```
import speech_recognition as sr
import pyttsx3
tasks = []
recognizer = sr.Recognizer()
engine = pyttsx3.init()
def add_task(task):
 tasks.append(task)
 engine.say(f"Task '{task}' added.")
 engine.runAndWait()
def view_tasks():
 if tasks:
     engine.say("Your tasks are:")
     for i, task in enumerate(tasks, start=1):
         engine.say(f"Task {i}: {task}")
 else:
     engine.say("No tasks to show.")
     engine.runAndWait()
def remove_task(task_number):
 if 0 < task_number <= len(tasks):
     removed_task = tasks.pop(task_number - 1)
     engine.say(f"Task '{removed_task}' removed.")
 else:
     engine.say("Invalid task number.")
     engine.runAndWait()
def recognize_speech():
 with sr.Microphone() as source:
     print("Listening...")
     audio = recognizer.listen(source)
     try:
         command = recognizer.recognize_google(audio)
         return command.lower()
     except sr.UnknownValueError:
         engine.say("Sorry, I did not understand that.")
         engine.runAndWait()
         return None
def main():
 while True:
     engine.say("Options: add task, view tasks, remove task, or exit.")
     engine.runAndWait()
     command = recognize_speech()
```