**EXERCISE 2: CLI, GUI AND VUI**

**Aim:**

To develop and compare Command Line Interface (CLI), Graphical User

Interface (GUI), and Voice User Interface (VUI) for the same task, and assess user

satisfaction using Python (with Tkinter for GUI and Speech Recognition for VUI)

and Terminal.

**Procedure:**

1. **CLI (Command-Line Interface)**

* **Set Up**:
  + Install Python and create a new script (e.g., todo.py).
* **Core Functions**:
  + add\_task(task): Adds a task.
  + view\_tasks(): Displays all tasks with numbering.
  + remove\_task(task\_number): Deletes a task by index.
  + save\_tasks(): Saves tasks to a file.
  + load\_tasks(): Loads tasks from a file at startup.
* **User Interaction**:
  + Use input() to capture user commands (add, view, remove, exit).
  + Implement a loop to continuously handle commands.
* **File Storage**:
  + Tasks are saved in tasks.txt.
  + Use file handling functions (open()) for reading and writing tasks.
* **Test & Run**:
  + Test the functions by running the script and checking task management functionality.

1. **GUI (Graphical User Interface)**

* **Set Up**:
  + Install Python, Tkinter (or PyQt6).
* **Create GUI Layout**:
  + Main window, entry box for task input, listbox for task display, buttons for add, remove, and clear actions.
* **Task Management Functions**:
  + add\_task(): Adds a task from the input box to the list.
  + remove\_task(): Deletes a selected task.
  + clear\_tasks(): Clears all tasks.
  + save\_tasks(): Saves tasks in a file.
  + load\_tasks(): Loads tasks at startup.
* **Event Handling**:
  + Bind buttons to respective functions and implement task removal on double-click.
* **Test & Run**:
  + Ensure all actions (add, remove, clear) work correctly in the GUI and run the script.

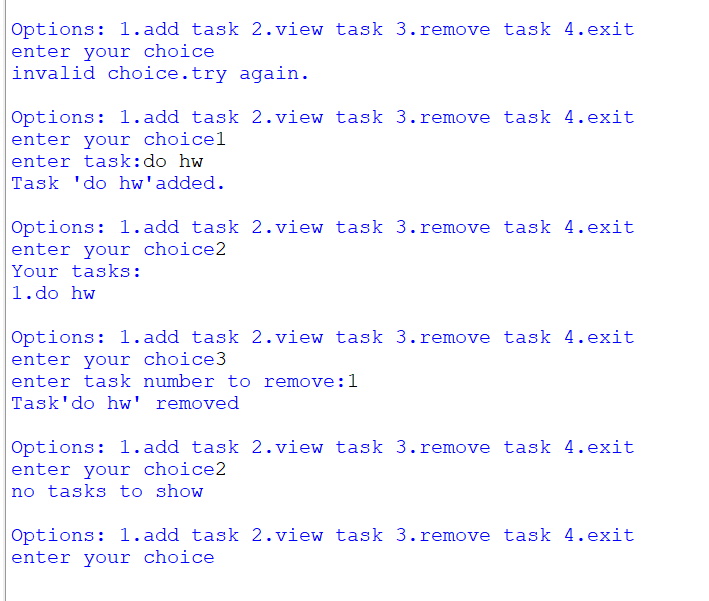
1. **VUI (Voice User Interface)**

* **Install Dependencies**:
  + Install necessary libraries: speechrecognition, pyttsx3, pyaudio.
* **Set Up Speech Recognition & Synthesis**:
  + Use speech\_recognition to convert speech to text.
  + Use pyttsx3 for text-to-speech responses.
* **Voice-Controlled Functions**:
  + listen\_command(): Captures voice input.
  + add\_task(task): Adds a task based on spoken input.
  + remove\_task(task\_number): Removes a task using voice input.
  + speak(text): Provides audio feedback.
* **Command Processing**:
  + Recognize voice commands (e.g., "Add task: Buy groceries" or "Remove task 2") and map them to functions.
* **Test & Improve**:
  + Test the script with different commands and refine speech recognition accuracy.

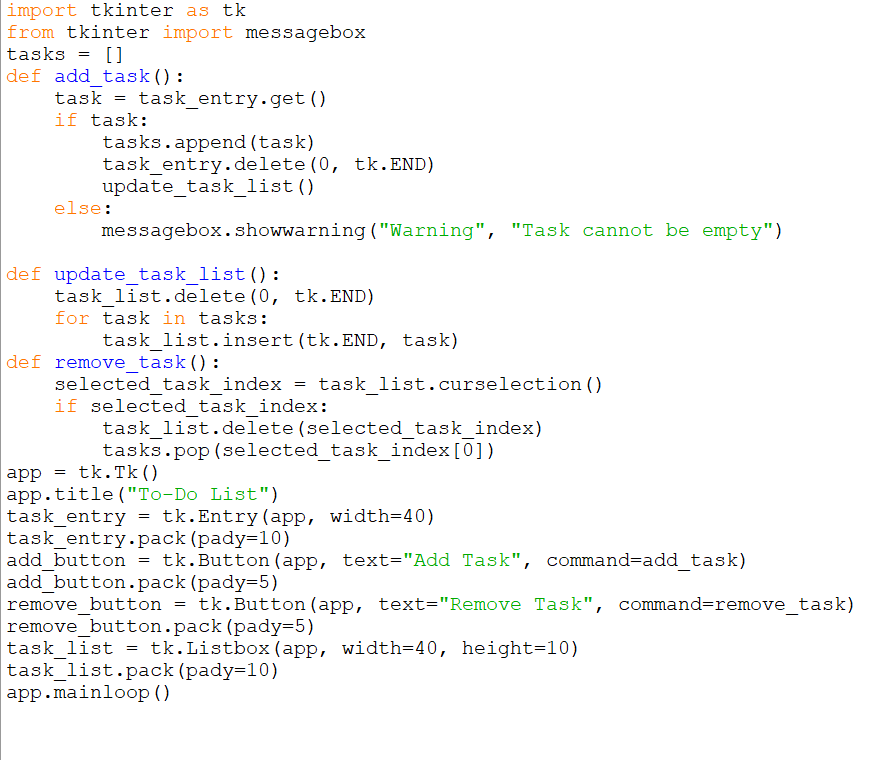
**Program & Output:**

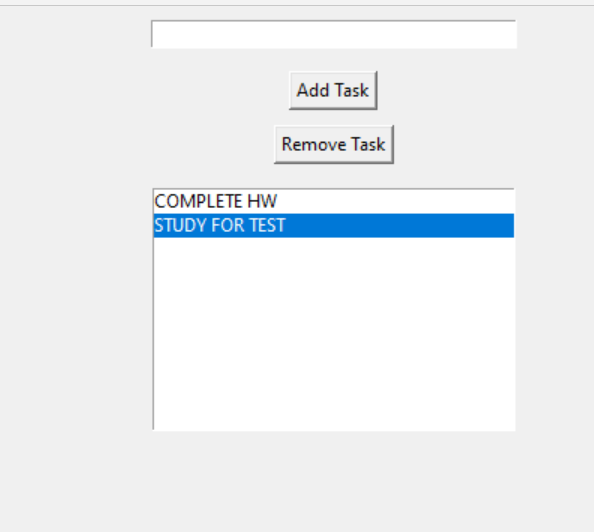
CLI:





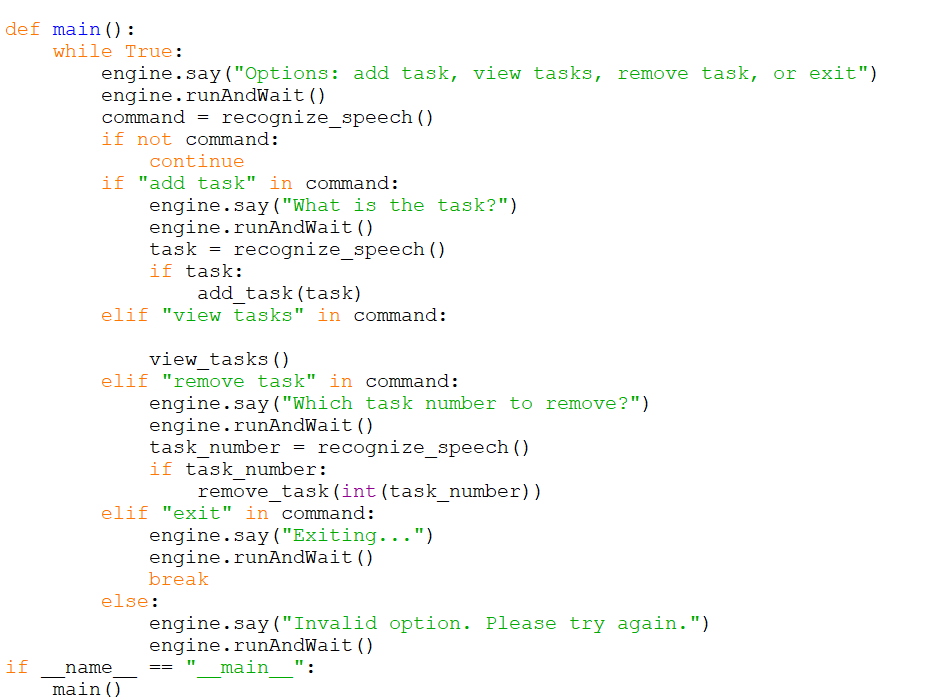
GUI:

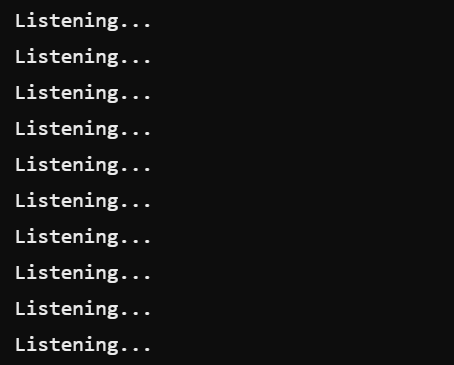




VUI:







Result:

The experiment successfully developed a To-Do application with CLI, GUI, and VUI interfaces, each allowing efficient task management and seamless user interaction.