NAME:THARSHINI M
CLASS:CSE F
ROLL NO:230701361

EXPERIMENT-3

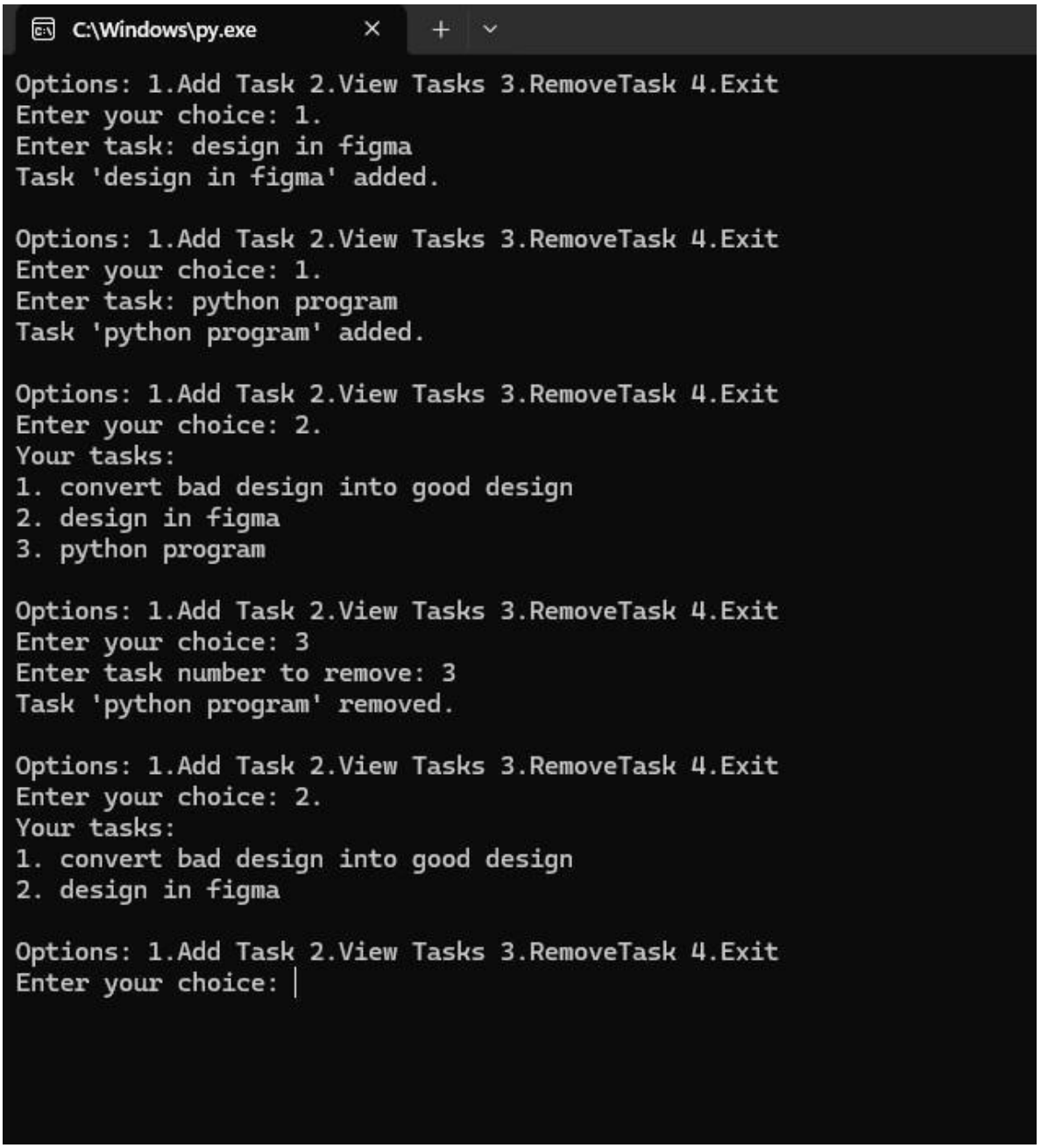
1.Develop and compare CLI, GUI, and Voice User Interfaces (VUI) for the same task and assess user satisfaction using Python (Tkinter for GUI, Speech Recognition for VUI), Terminal AIM: The aim is to develop and compare Command Line Interface (CLI), Graphical User Interface (GUI), and Voice User Interface (VUI) for the same task, and assess user satisfaction using Python (with Tkinter for GUI and Speech Recognition for VUI) and Terminal. PROCEDURE:

A)Command Line Interface

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

def view_tasks():
    if tasks:
        print("Your tasks:")
        for idx, task in enumerate(tasks, 1):
```

```
            print(f"{idx}. {task}")
    else:
        print("No tasks to show.")

def remove_task(task_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.")
```

```
while True:
        print("\nOptions: 1.Add Task 2.View Tasks 3.RemoveTask 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':
            task_number = int(input("Enter task number to remove: "))
            remove_task(task_number)
        elif choice == '4':
            print("Exiting...")
            break
        else:
            print("Invalid choice. Please try again.")

if __name__== "__main__":
    main()
```

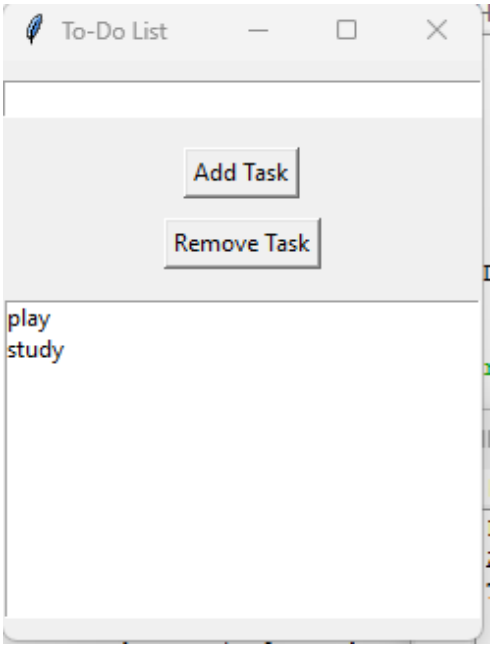

C:\Windows\py.exe
Options: 1.Add Task 2.View Tasks 3.RemoveTask 4.Exit
Enter your choice: 1.
Enter task: design in figma
Task 'design in figma' added.
Options: 1.Add Task 2.View Tasks 3.RemoveTask 4.Exit
Enter your choice: 1.
Enter task: python program
Task 'python program' added.
Options: 1.Add Task 2.View Tasks 3.RemoveTask 4.Exit
Enter your choice: 2.
Your tasks:
1. convert bad design into good design
2. design in figma
3. python program
Options: 1.Add Task 2.View Tasks 3.RemoveTask 4.Exit
Enter your choice: 3
Enter task number to remove: 3
Task 'python program' removed.
Options: 1.Add Task 2.View Tasks 3.RemoveTask 4.Exit
Enter your choice: 2.
Your tasks:
1. convert bad design into good design
2. design in figma
Options: 1.Add Task 2.View Tasks 3.RemoveTask 4.Exit
Enter your choice:

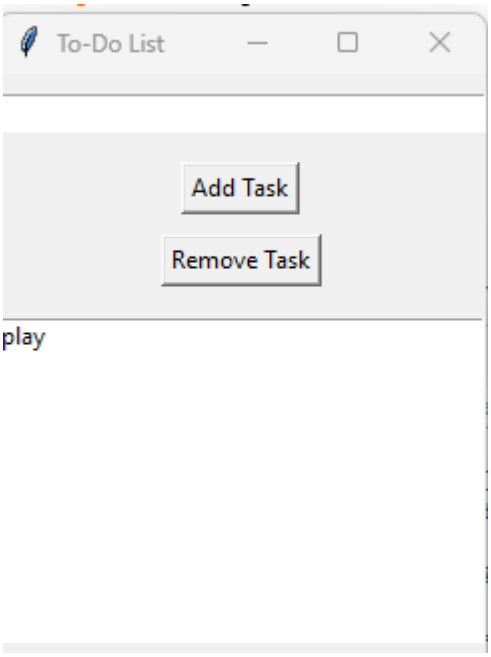
b) Graphical User Interface

```
import tkinter as tk
from tkinter import messagebox
tasks = []
def add_task():
    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():
    task_list.delete(0, tk.END)
    for task in tasks:
        task_list.insert(tk.END, task)
```

```
def remove_task():
    selected_task_index = task_list.curselection()
    if selected_task_index:
        task_list.delete(selected_task_index)
        tasks.pop(selected_task_index[0])

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()
```





c) Voice user interface

```
import
speech_recognition
as sr import os


def
 rename_file_from_voice_co
 mmand(command): try:
   words = command.lower().split(" ")
   if "rename" in words and "to" in words:
     rename_index =
```

```
words.index("rename"
) to_index =
words.index("to")

# Extract old and new filenames
old_name =
words[rename_index
+ 1] new_name =
words[to_index + 1]

# Check if file exists
if not os.path.exists(old_name):
  print(f"Error: File
  '{old_name}' not found.")
  return
```

```
        # Rename file
        os.rename(old_name, new_name)
        print(f"✅ File renamed from '{old_name}' to '{new_name}'")

    else:
        print("Invalid command format. Say: 'Rename oldfile.txt to newfile.txt'")

    except Exception as e:
        print(f"Error: {e}")

def listen_for_command():
    recognizer = sr.Recognizer()
    mic = sr.Microphone()

    print("🎤 Listening for
```

```
command to rename a file...")
with mic as source:
  recognizer.adjust_for_amb
  ient_noise(source) audio =
  recognizer.listen(source)

try:
  command =
  recognizer.recognize_google(aud
  io) print(f" 🎙️ Command
  received: {command}")
  rename_file_from_voice_c
ommand(command) except
sr.UnknownValueError:
  print(" ❌ Sorry, I couldn't
understand the command.")
except sr.RequestError as e:
  print(f" ⚠️ Could not request results from Google
  Speech Recognition service; {e}")
```

```
if _name____==
  "_main_":
  listen_for_co
  mmand()
```

```
D:\230701304>python rename_vui.py
🎤 Listening for command to rename a file...
🎙️ Command received: rename
Invalid command format. Say: 'Rename oldfile.txt to newfile.txt'

D:\230701304>python rename_vui.py
🎤 Listening for command to rename a file...
❌ Sorry, I couldn't understand the command.
```

```
[Running] python -u "d:\230701304\vui.py"
Listening for command to rename a file...
Command received: rename sample to Shark Shal
File renamed from sample to Shark
```