CS23A34

USER INTERFACE DESIGN

EXPERIMENT-3

ROLL NO: 240701256

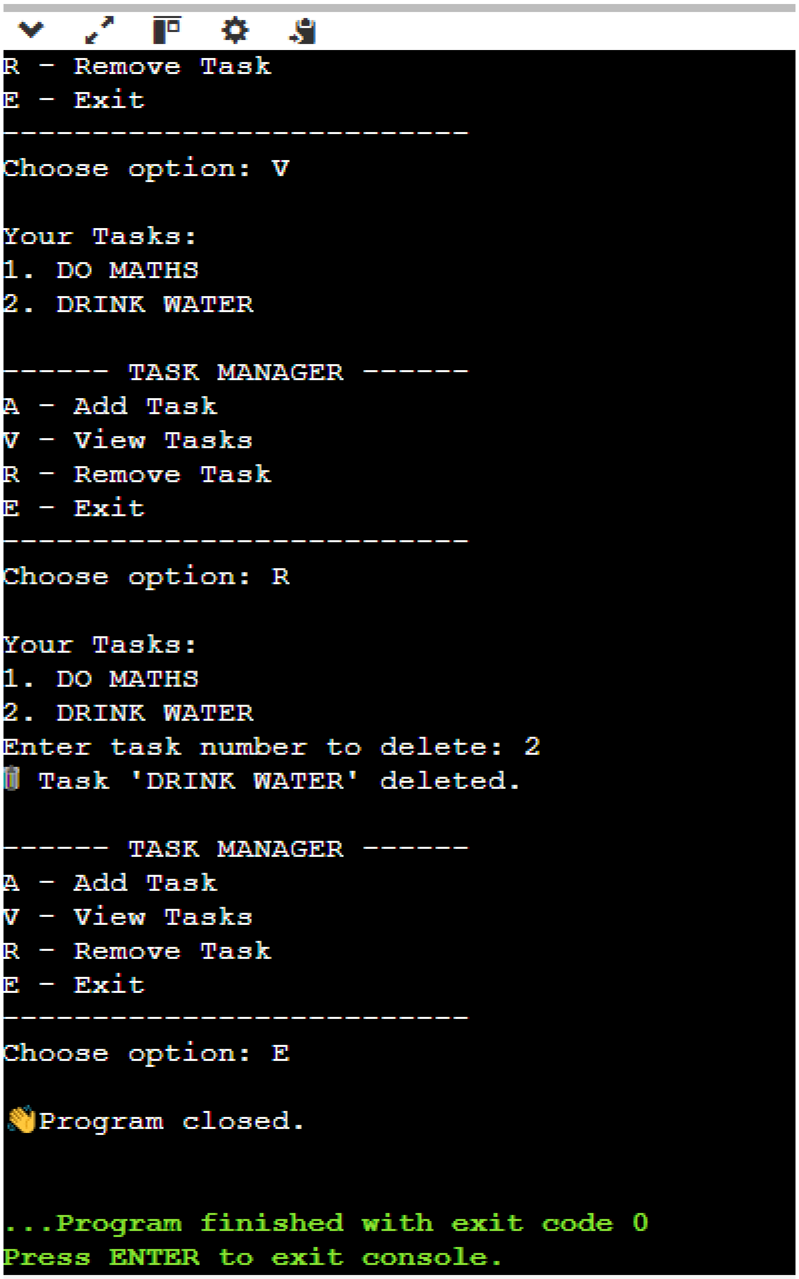
NAME: KEERTHANA R

1. COMMAND LINE INTERFACE

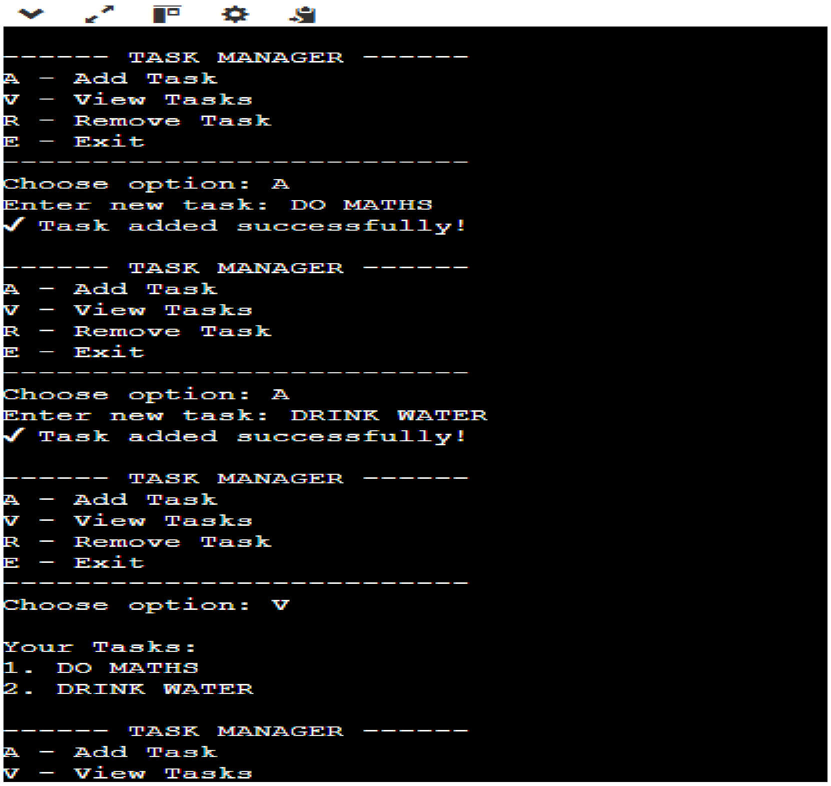
PYTHON CODE :

```
# Simple CLI Task Manager

task_list = []

def menu():
    print("\n------ TASK MANAGER ------")
    print("A - Add Task")
    print("V - View Tasks")
    print("R - Remove Task")
    print("E - Exit")
    print("--------------------------")

def add():
    task = input("Enter new task: ")
    task_list.append(task)
    print("Task added successfully!")

def view():
    if len(task_list) == 0:
        print(" No tasks found.")
    else:
        print("\nYour Tasks:")
        for i in range(len(task_list)):
            print(f"{i+1}. {task_list[i]}")

def remove():
    view()
    if len(task_list) == 0:
        return
    try:
        num = int(input("Enter task number to delete: "))
        if num >= 1 and num <= len(task_list):
            deleted = task_list.pop(num-1)
            print(f"Task '{deleted}' deleted.")
        else:
            print(“ Invalid number.")
    except:
        print("Please enter a valid integer.")

def run():
    while True:
        menu()
        choice = input("Choose option: ").upper()

        if choice == "A":
            add()
        elif choice == "V":
            view()
        elif choice == "R":
            remove()
        elif choice == "E":
            print("\n Program closed.")
            break
        else:
```

```
        print("Invalid option. Try again.")

# Start program
run()
```



```
R - Remove Task
E - Exit
--------------------------
Choose option: V

Your Tasks:
1. DO MATHS
2. DRINK WATER

------ TASK MANAGER ------
A - Add Task
V - View Tasks
R - Remove Task
E - Exit
--------------------------
Choose option: R

Your Tasks:
1. DO MATHS
2. DRINK WATER
Enter task number to delete: 2
🗑 Task 'DRINK WATER' deleted.

------ TASK MANAGER ------
A - Add Task
V - View Tasks
R - Remove Task
E - Exit
--------------------------
Choose option: E

👋Program closed.


...Program finished with exit code 0
Press ENTER to exit console.
```



```
------ TASK MANAGER ------
A - Add Task
V - View Tasks
R - Remove Task
E - Exit
--------------------------
Choose option: A
Enter new task: DO MATHS
✓ Task added successfully!

------ TASK MANAGER ------
A - Add Task
V - View Tasks
R - Remove Task
E - Exit
--------------------------
Choose option: A
Enter new task: DRINK WATER
✓ Task added successfully!

------ TASK MANAGER ------
A - Add Task
V - View Tasks
R - Remove Task
E - Exit
--------------------------
Choose option: V

Your Tasks:
1. DO MATHS
2. DRINK WATER

------ TASK MANAGER ------
A - Add Task
V - View Tasks
```

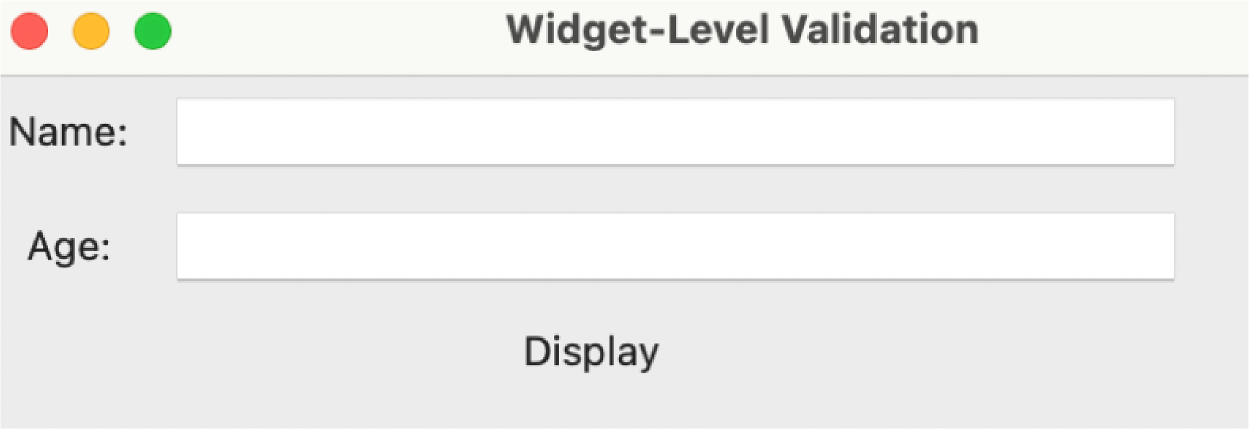
2.GRAPHICAL USER INTERFACE:

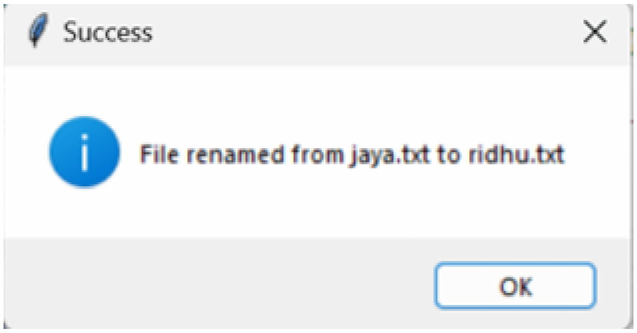
```
import tkinter as tk

from tkinter import messagebox


# Function to submit data

def submit_data():

    name = name_entry.get()

    age = age_entry.get()

    dept = dept_entry.get()
    if name == "" or age == "" or dept == "":
        messagebox.showwarning("Warning", "All fields are required!")
    else:
        result_label.config(
            text=f"Student Registered!\n\nName: {name}\nAge: {age}\nDepartment:
{dept}"
        )
        clear_fields()

# Function to clear fields
def clear_fields():
    name_entry.delete(0, tk.END)
    age_entry.delete(0, tk.END)
    dept_entry.delete(0, tk.END)

# Create main window
window = tk.Tk()
window.title("Student Registration - GUI")
window.geometry("400x350")
window.resizable(False, False)

# Title
title_label = tk.Label(window, text="Student Registration Form",
                       font=("Arial", 16, "bold"))
```

```
title_label.pack(pady=15)

# Name
tk.Label(window, text="Name:").pack()
name_entry = tk.Entry(window, width=30)
name_entry.pack(pady=5)

# Age
tk.Label(window, text="Age:").pack()
age_entry = tk.Entry(window, width=30)
age_entry.pack(pady=5)

# Department
tk.Label(window, text="Department:").pack()
dept_entry = tk.Entry(window, width=30)
dept_entry.pack(pady=5)

# Buttons
tk.Button(window, text="Submit", width=15, command=submit_data).pack(pady=10)
tk.Button(window, text="Clear", width=15, command=clear_fields).pack()

# Result Label
result_label = tk.Label(window, text="", font=("Arial", 11))
result_label.pack(pady=15)
```

```
# Run window

window.mainloop()
```

output:





File renamed from jaya.txt to ridhu.txt

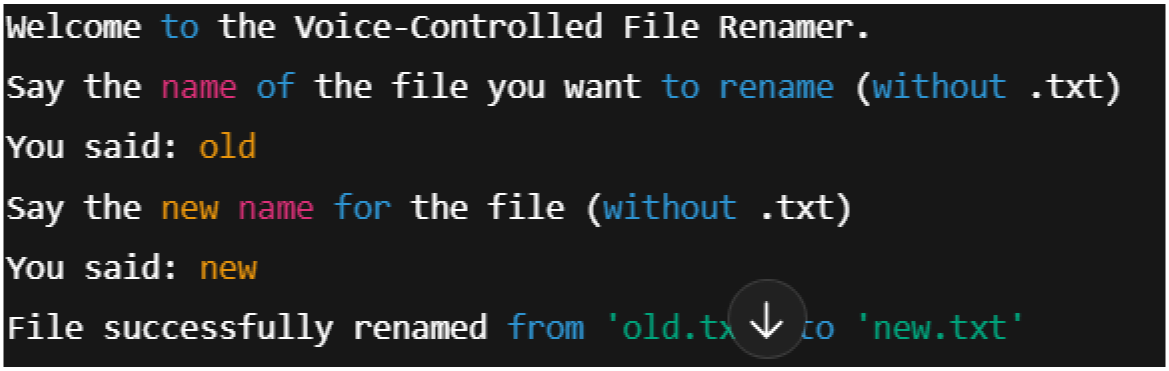
3.Voice user interface(VUI)

```
    import speech_recognition as sr

import pyttsx3


# Initialize recognizer and text-to-speech engine

recognizer = sr.Recognizer()
```

```python
engine = pyttsx3.init()


def speak(text):
    engine.say(text)
    engine.runAndWait()


def listen():
    with sr.Microphone() as source:
        print("Speak something...")
        recognizer.adjust_for_ambient_noise(source)
        audio = recognizer.listen(source)


        try:
            text = recognizer.recognize_google(audio)
            print("You said:", text)
            speak("You said " + text)
        except sr.UnknownValueError:
            print("Sorry, I could not understand.")
            speak("Sorry, I could not understand.")
        except sr.RequestError:
            print("Network error.")
            speak("Network error.")


# Run the VUI
```

```
speak("Voice system started. Please speak.")

listen()
```

output :



```
Welcome to the Voice-Controlled File Renamer.
Say the name of the file you want to rename (without .txt)
You said: old
Say the new name for the file (without .txt)
You said: new
File successfully renamed from 'old.tx ↓ to 'new.txt'
```