EXPERIMENT 3-USER INTERFACE DESIGN

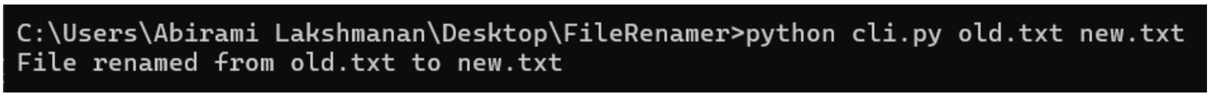
Abirami L-240701010

Command Line Interface:

PYTHON CODE:

```
import os

import sys


def rename_file(old_name, new_name):

    try:

        os.rename(old_name, new_name)

        print(f"File renamed from {old_name} to {new_name}")

    except FileNotFoundError:

        print(f"Error: {old_name} not found.")

    except Exception as e:

        print(f"An error occurred: {e}")


if __name__ == "__main__":

    if len(sys.argv) != 3:

        print("Usage: python rename_file_cli.py <old_filename> <new_filename>")

    else:

        rename_file(sys.argv[1], sys.argv[2])
```

OUTPUT:



```
C:\Users\Abirami Lakshmanan\Desktop\FileRenamer>python cli.py old.txt new.txt
File renamed from old.txt to new.txt
```

Graphical User Interface:

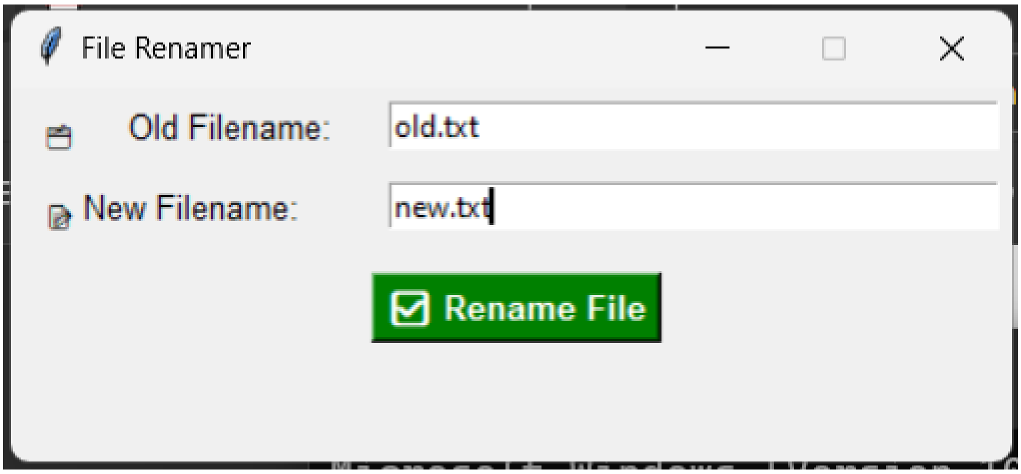
```
import tkinter as tk

from tkinter import messagebox

import os
```

```
# Function to rename the file
def rename_file():
    old_name = old_filename_entry.get().strip()
    new_name = new_filename_entry.get().strip()

    # Check if input fields are empty
    if not old_name or not new_name:
        messagebox.showwarning("Warning", "Please enter both filenames!")
        return

    # Check if the old file exists
    if not os.path.exists(old_name):
        messagebox.showerror("Error", f"File '{old_name}' not found.")
        return

    # Check if the new file already exists
    if os.path.exists(new_name):
        overwrite = messagebox.askyesno(
            "Warning", f"'{new_name}' already exists. Overwrite?"
        )
        if not overwrite:
            return

    try:
        os.rename(old_name, new_name)
        messagebox.showinfo(
            "Success", f"File renamed from '{old_name}' to '{new_name}'"
        )
    except Exception as e:
        messagebox.showerror("Error", f"An error occurred: {e}")
```

```
# Create main window
root = tk.Tk()
root.title("File Renamer")
root.geometry("400x150")
root.resizable(False, False)

# Labels
tk.Label(root, text="🗂️ Old Filename:", font=("Arial", 10)).grid(
    row=0, column=0, padx=10, pady=5, sticky="w"
)
tk.Label(root, text="📝 New Filename:", font=("Arial", 10)).grid(
    row=1, column=0, padx=10, pady=5, sticky="w"
)

# Entry fields
old_filename_entry = tk.Entry(root, width=40)
old_filename_entry.grid(row=0, column=1, padx=10, pady=5)

new_filename_entry = tk.Entry(root, width=40)
new_filename_entry.grid(row=1, column=1, padx=10, pady=5)

# Styled Button
rename_button = tk.Button(
    root,
    text="✅ Rename File",
    bg="green",
    fg="white",
    font=("Arial", 10, "bold"),
    command=rename_file
)
```

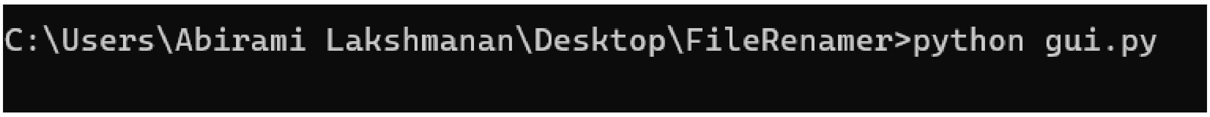
```
rename_button.grid(row=2, column=0, columnspan=2, pady=10)


# Run the GUI event loop
root.mainloop()
```

OUTPUT:

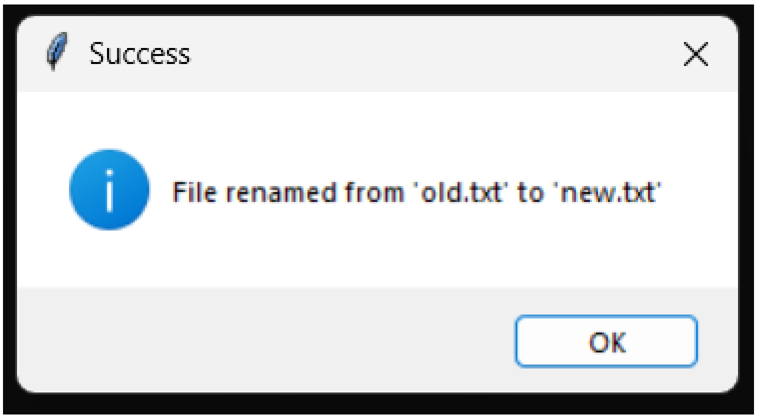


Old Filename: old.txt

New Filename: new.txt



```
C:\Users\Abirami Lakshmanan\Desktop\FileRenamer>python gui.py
```

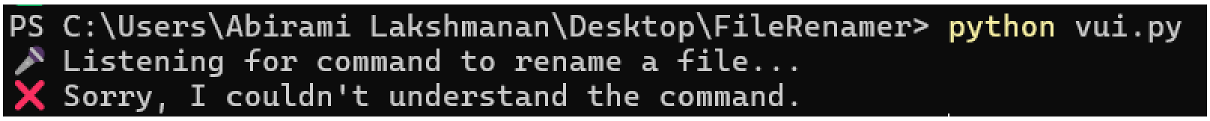


File renamed from 'old.txt' to 'new.txt'

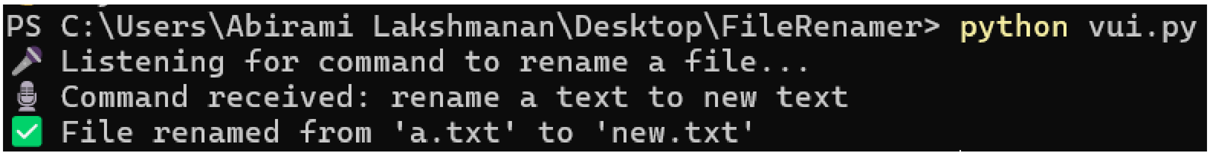
Voice User Interface:

```
import speech_recognition as sr
import os


def rename_file_from_voice_command(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.")
            print("👉 Say: Rename oldfile.txt to newfile.txt")

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

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

    print("🎤 Listening for command to rename a file...")

    with microphone as source:
        recognizer.adjust_for_ambient_noise(source)

        audio = recognizer.listen(source)


    try:

        command = recognizer.recognize_google(audio)

        print(f"🎙️ Command received: {command}")

        rename_file_from_voice_command(command)


    except sr.UnknownValueError:

        print("❌ Sorry, I couldn't understand the command.")

    except sr.RequestError as e:

        print(f"⚠️ Speech recognition service error: {e}")


if __name__ == "__main__":

    listen_for_command()
```

OUTPUT:



```
PS C:\Users\Abirami Lakshmanan\Desktop\FileRenamer> python vui.py
🎤 Listening for command to rename a file...
❌ Sorry, I couldn't understand the command.
```

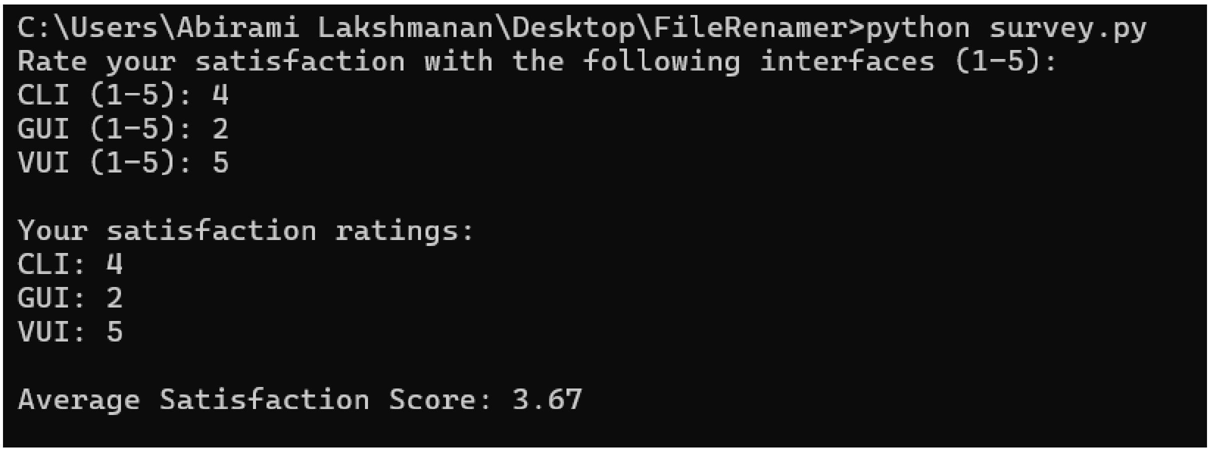


```
PS C:\Users\Abirami Lakshmanan\Desktop\FileRenamer> python vui.py
🎤 Listening for command to rename a file...
🎙️ Command received: rename a text to new text
✅ File renamed from 'a.txt' to 'new.txt'
```

User Satisfaction Comparison:

```
def survey():

    print("Rate your satisfaction with the following interfaces (1-5):")


    try:

        # Get user input

        cli_satisfaction = int(input("CLI (1-5): "))

        gui_satisfaction = int(input("GUI (1-5): "))

        vui_satisfaction = int(input("VUI (1-5): "))
        # Ensure valid ratings
        if not (1 <= cli_satisfaction <= 5 and
                1 <= gui_satisfaction <= 5 and
                1 <= vui_satisfaction <= 5):
            print("Please enter ratings between 1 and 5 only.")
            return

        # Display the ratings
        print("\nYour satisfaction ratings:")
        print(f"CLI: {cli_satisfaction}")
        print(f"GUI: {gui_satisfaction}")
        print(f"VUI: {vui_satisfaction}")

        # Calculate average satisfaction
        avg_satisfaction = (
            cli_satisfaction + gui_satisfaction + vui_satisfaction
        ) / 3

        print(f"\nAverage Satisfaction Score: {avg_satisfaction:.2f}")

    except ValueError:
        print("Invalid input! Please enter numbers between 1 and 5.")

# Run the survey function
if __name__ == "__main__":
    survey()
```

OUTPUT:



```
C:\Users\Abirami Lakshmanan\Desktop\FileRenamer>python survey.py
Rate your satisfaction with the following interfaces (1-5):
CLI (1-5): 4
GUI (1-5): 2
VUI (1-5): 5

Your satisfaction ratings:
CLI: 4
GUI: 2
VUI: 5

Average Satisfaction Score: 3.67
```