Exp No: 2 UID -2

230701248 Date:08.02.2025

Difference between Command Line Interface, Graphical User Interface and Voice User Interface

CLI

- Simple and lightweight command-line script.
- Rename files using terminal commands.
- Ideal for developers and automation scripts.

Program:

```
import os
import sys

def rename_file(old_name, new_name):
    try:
        os.rename(old_name, new_name)  # Use function parameters
        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__":  # Fixed the main check
    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:

```
File renamed from filetoberenamed.txt to filetoberenamed2.txt
```

GUI:

- User-friendly interface using Tkinter.
- Allows easy input of filenames.
- Ideal for non-technical users.

Program:

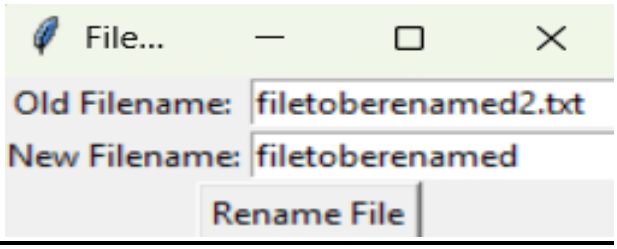
```
import tkinter as tk
from tkinter import messagebox
import os

def rename_file():
    old_name = old_filename_entry.get()
    new_name = new_filename_entry.get()

    try:
        os.rename(old_name, new_name)
        messagebox.showinfo("Success", f"File renamed from {old_name} to {new_name}")
    except FileNotFoundError:
        messagebox.showerror("Error", f"File {old_name} not found.")
    except Exception as e:
        messagebox.showerror("Error", f"An error occurred: {e}")

root = tk.Tk()
root.title("File Renamer")

tk.Label(root, text="Old Filename:").grid(row=0, column=0)
tk.Label(root, text="New Filename:").grid(row=1, column=0)

old_filename_entry = tk.Entry(root)
old_filename_entry.grid(row=0, column=1)

new_filename_entry = tk.Entry(root)
new_filename_entry.grid(row=1, column=1)

rename_button = tk.Button(root, text="Rename File", command=rename_file)
rename_button.grid(row=2, columnspan=2)

root.mainloop()
```

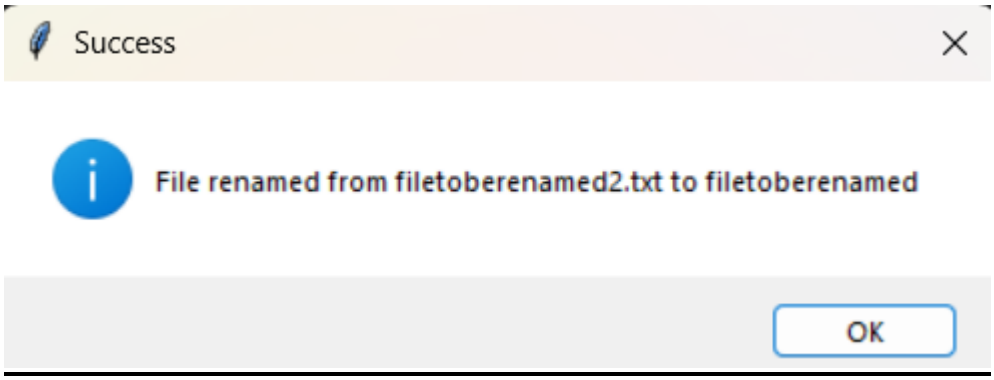
Command:

```
D:\Downloads\My Apps(Built Ones)\college>python gui.py
```

Output:

Output:





VUI:

- Uses Speech Recognition to rename files.
- Hands-free operation using voice commands.
- Suitable for accessibility and innovative automation.

Program:

```
import speech_recognition as sr
import os

def rename_file_from_voice_command(old_name, new_name):
    """Renames a file using the provided old and new names."""
    try:
        old_name += ".txt"
        new_name += ".txt"

        if not os.path.exists(old_name):
            print(f"❌ Error: '{old_name}' not found.")
            return
                                (parameter) new_name: Any
        os.rename(old_name, new_name)
        print(f"✅ File successfully renamed from '{old_name}' to '{new_name}'")
    except Exception as e:
        print(f"❌ Error: {e}")

def listen_for_filename(prompt):
    """Listens for a single filename input via voice command."""
    recognizer = sr.Recognizer()
    mic = sr.Microphone()

    with mic as source:
        recognizer.adjust_for_ambient_noise(source, duration=3)  # Increase noise adaptation
        print(f"🎤 {prompt}")
        try:
            audio = recognizer.listen(source, timeout=10, phrase_time_limit=5)  # Increased timeout
            command = recognizer.recognize_google(audio, language="en-US")
            print(f"📝 You said: {command}")
            return command.strip().replace(" ", "_")  # Replace spaces with underscores
        except sr.UnknownValueError:
            print("❌ Could not understand. Please try again.")
            return None
        except sr.WaitTimeoutError:
            print("⏳ Timeout: No speech detected. Try speaking louder and clearly.")
```

```
        except sr.RequestError as e:
            print(f"❌ Error with speech recognition service: {e}")
            return None

if __name__ == "__main__":
    print("🔊 Welcome to the Voice-Controlled File Renamer!")

    old_name = None
    while old_name is None:
        old_name = listen_for_filename("Say the name of the file you want to rename (without .txt)")

    new_name = None
    while new_name is None:
        new_name = listen_for_filename("Say the new name for the file (without .txt)")

    rename_file_from_voice_command(old_name, new_name)
```

Command:

```
d:\Downloads\My Apps(Built Ones)\college>python vui.py
```

Output:

```
d:\Downloads\My Apps(Built Ones)\college> d: && cd "d:\Downloads\My Apps(Built Ones)\college" && cmd /C ""c:\Program Files\Python313\python.exe" c:\Users\LENOVO\.vscode\extensions\ms-python.debug
py-2025.0.0-win32-x64\bundled\libs\debugpy\launcher 55641 -- "d:\Downloads\My Apps(Built Ones)\college\vui.py" "
Welcome to the Voice-Controlled File Renamer!
Say the name of the file you want to rename (without .txt)
You said: new file
Say the new name for the file (without .txt)
You said: old
File successfully renamed from 'new_file.txt' to 'old.txt'
```