UID EX 3

Rama Thulasi G
230701259

Command Line Interface (CLI)

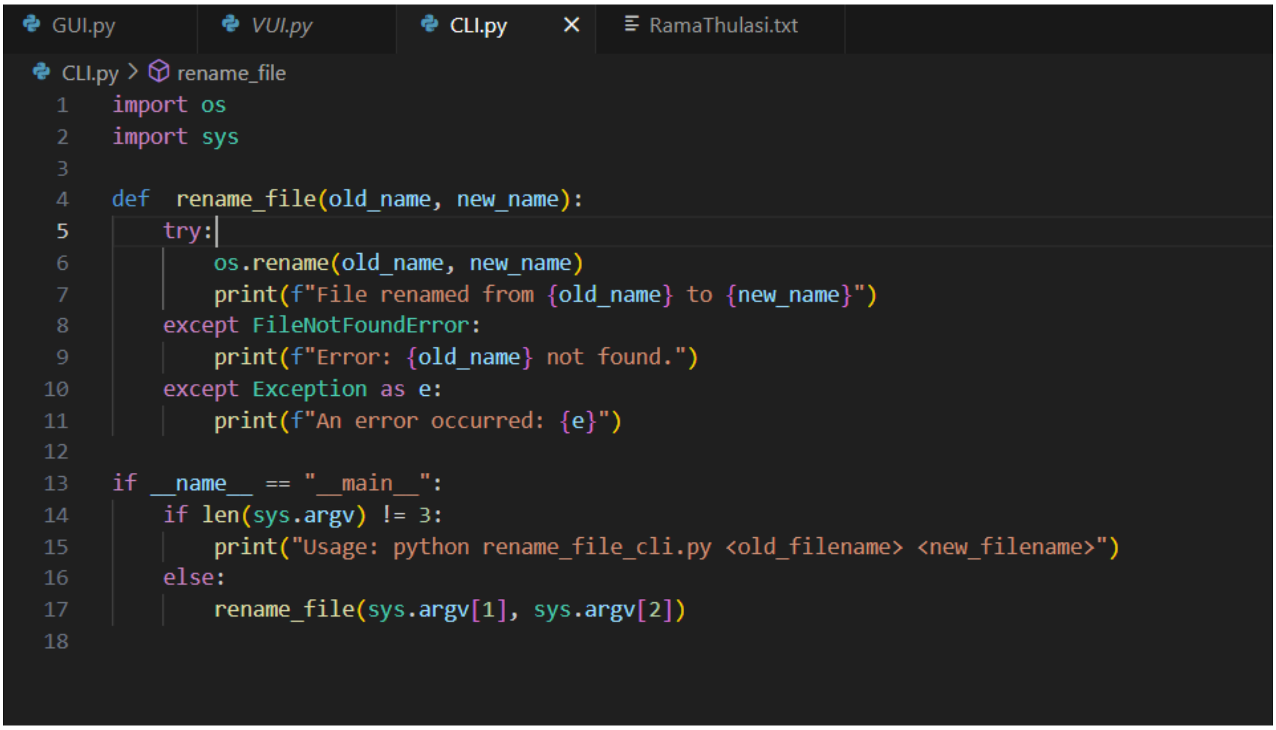
- A **Command Line Interface (CLI)** is a text-based interface where users interact with a computer by typing commands.
- It is commonly used by developers, system administrators, and advanced users to perform tasks efficiently.
- Very fast for experienced users who know commands.

Advantages:

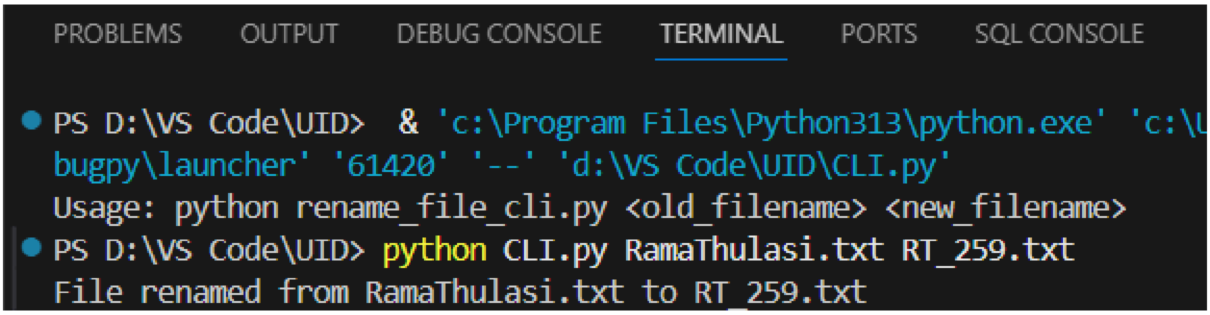
- Faster for experienced users
- Uses fewer system resources

Disadvantages:

- Requires memorizing commands
- No visual elements



```python
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])

```



```
● PS D:\VS Code\UID>  & 'c:\Program Files\Python313\python.exe' 'c:\U
  bugpy\launcher' '61420' '--' 'd:\VS Code\UID\CLI.py'
  Usage: python rename_file_cli.py <old_filename> <new_filename>
● PS D:\VS Code\UID> python CLI.py RamaThulasi.txt RT_259.txt
  File renamed from RamaThulasi.txt to RT_259.txt
```

Graphical User Interface (GUI)

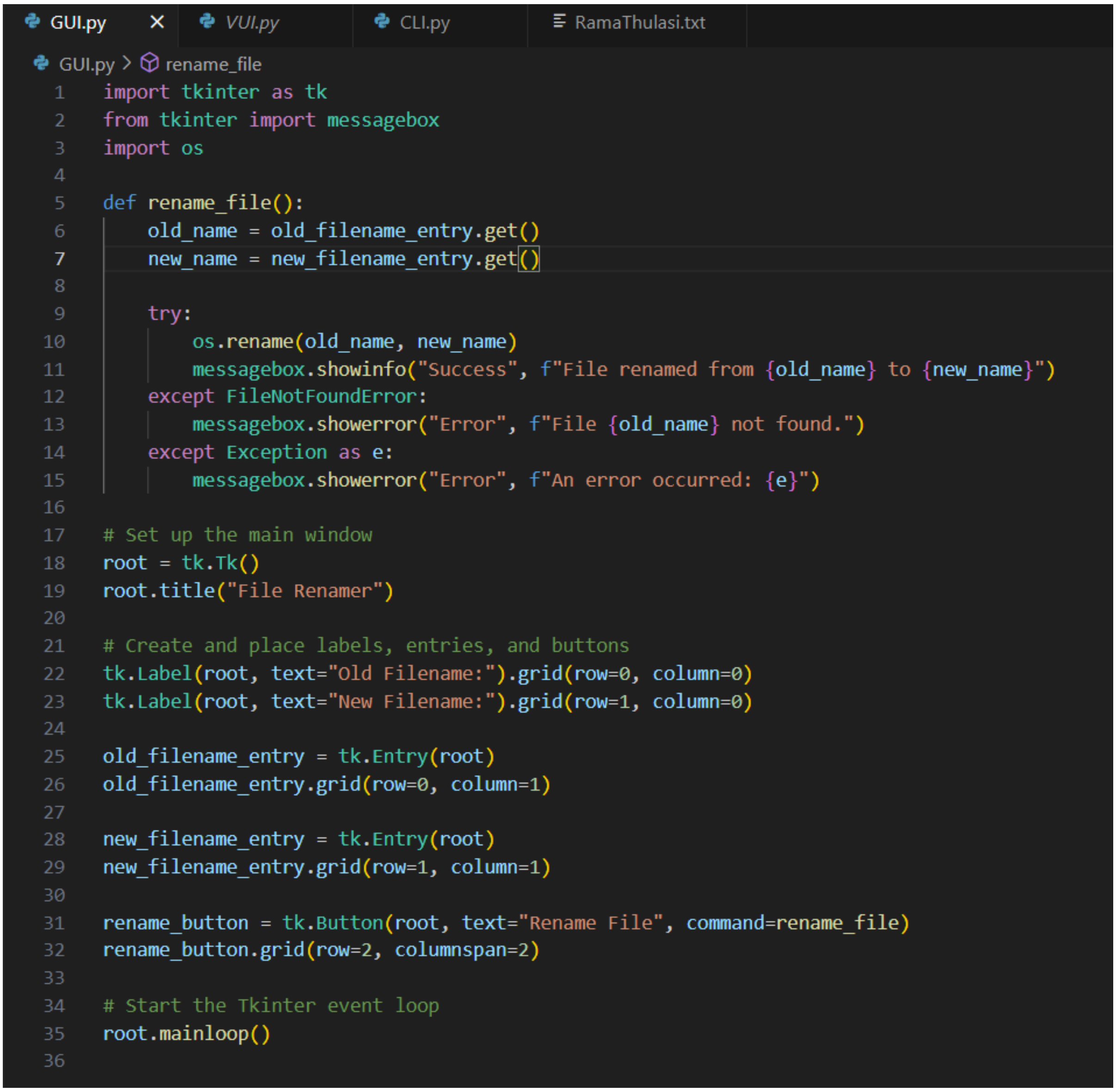
- A **Graphical User Interface (GUI)** allows users to interact with a computer using visual elements such as windows, buttons, icons, and menus.
- It is designed for ease of use, making computers accessible to non-technical users.

Advantages:

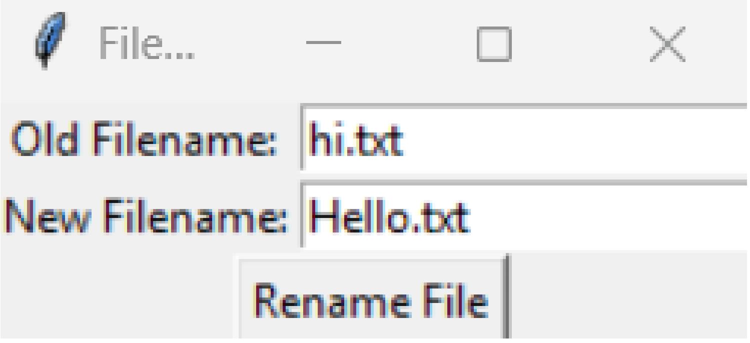
- Easy to use, beginner-friendly
- Does not require memorizing commands
- More interactive with visuals

Disadvantages:

- Slower than CLI for advanced users
- Requires more system resources
- Limited automation capabilities

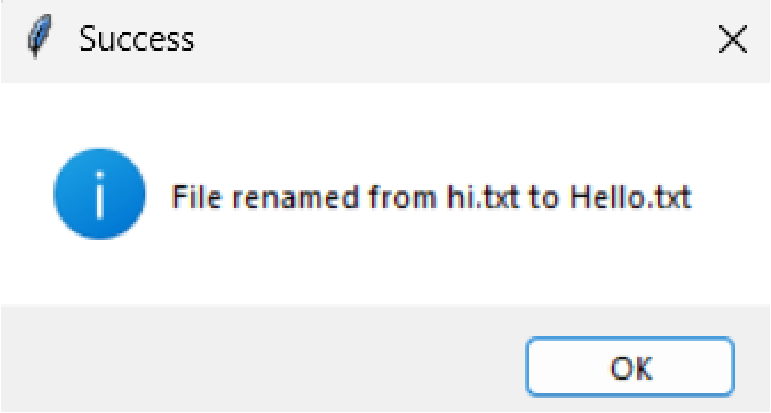


```python
import tkinter as tk
from tkinter import messagebox
import os

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}")

# Set up the main window
root = tk.Tk()
root.title("File Renamer")

# Create and place labels, entries, and buttons
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)

# Start the Tkinter event loop
root.mainloop()
```



Old Filename: hi.txt

New Filename: Hello.txt



Success

File renamed from hi.txt to Hello.txt

Voice User Interface (VUI)

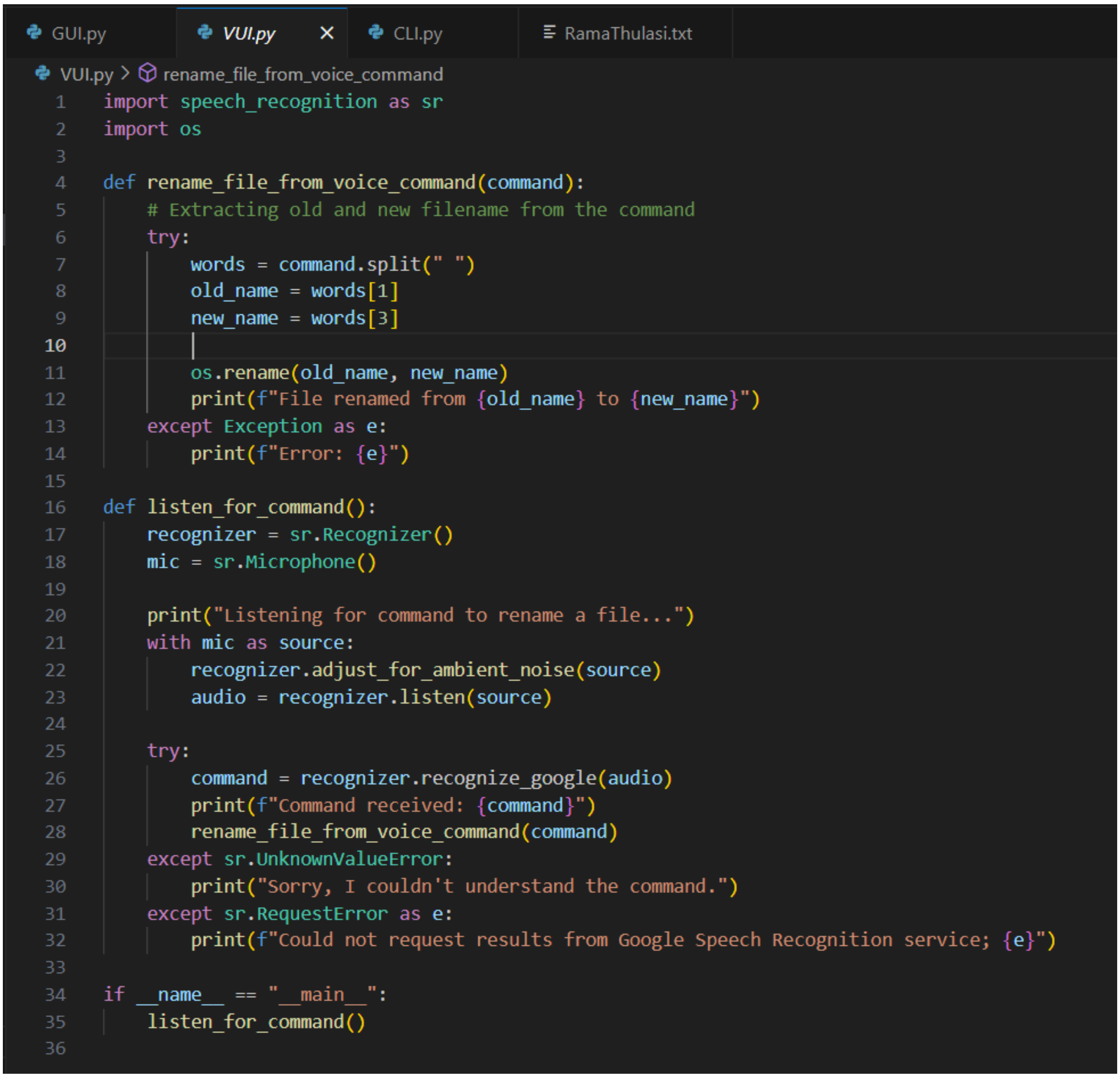
- A **Voice User Interface (VUI)** allows users to interact with a computer or device using spoken commands.
- It is commonly used in smart assistants, voice-controlled applications, and accessibility features.

Advantages:

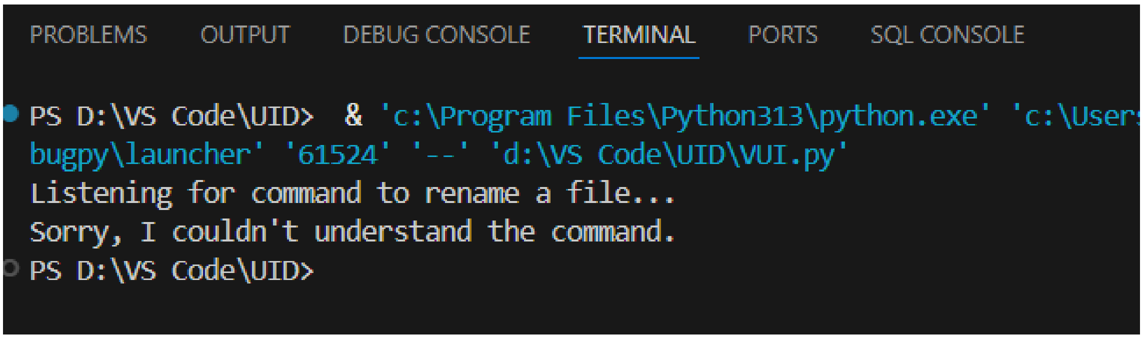
- Hands-free and convenient
- Easy to use for all ages
- Useful for people with disabilities

Disadvantages:

- Accuracy depends on speech recognition
- Struggles with different accents and background noise
- Limited to predefined commands



```python
import speech_recognition as sr
import os

def rename_file_from_voice_command(command):
    # Extracting old and new filename from the command
    try:
        words = command.split(" ")
        old_name = words[1]
        new_name = words[3]

        os.rename(old_name, new_name)
        print(f"File renamed from {old_name} to {new_name}")
    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_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"Could not request results from Google Speech Recognition service; {e}")

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



```
PS D:\VS Code\UID>  & 'c:\Program Files\Python313\python.exe' 'c:\Users
bugpy\launcher' '61524' '--' 'd:\VS Code\UID\VUI.py'
Listening for command to rename a file...
Sorry, I couldn't understand the command.
PS D:\VS Code\UID>
```