Explanation of Continuous Keylogger Project

# 1. dashboard\_continuous.py

**#!/usr/bin/env python3**

Shebang line that tells the system to use Python 3 interpreter when running the script.

**import os, time, threading, socket, datetime, pynput.keyboard, colorama**

Imports required modules: OS for folder handling, time for delays, threading for background server, socket for local networking, datetime for timestamps, pynput for capturing keys, colorama for colored terminal output.

**init(autoreset=True)**

Initializes colorama to allow colored terminal printing with auto-reset.

**HOST = '127.0.0.1'; PORT = 65432**

Defines host and port for local socket communication (loopback address).

**def start\_server():**

Function to start a background server thread that listens for keystroke messages.

**server\_loop()**

Defines loop that accepts socket connections and logs received keystrokes to server\_log.txt.

**def send\_keystroke(key\_text):**

Tries to send a keystroke over socket to the local server; fails silently if server not ready.

**def create\_session\_folder():**

Creates a timestamped folder for each logging session.

**def save\_session(folder\_name, keystrokes, stats):**

Writes keystrokes and summary statistics to text files inside the session folder.

**def display\_dashboard(stats):**

Clears the terminal and displays an ASCII dashboard with live statistics of letters, numbers, etc.

**def run\_single\_session():**

Captures one full session of keystrokes until user presses Esc.

**keyboard.Listener(on\_press=on\_press)**

Listens for key events and calls on\_press callback for each key.

**on\_press(key)**

Handles each keystroke, classifies it, updates stats, sends via socket, updates dashboard, stops if Esc pressed.

**def main\_loop():**

Main program loop: starts the server once, then repeatedly runs sessions, waits 2 seconds between them, stops on Ctrl+C.

**if \_\_name\_\_ == '\_\_main\_\_': main\_loop()**

Entry point of script.

# 2. simple\_continuous.py

**#!/usr/bin/env python3**

Shebang line for Python 3 interpreter.

**import os, time, threading, socket, datetime, pynput.keyboard**

Imports modules: OS, time, threading, socket, datetime, and pynput for capturing keystrokes.

**HOST = '127.0.0.1'; PORT = 65432**

Local socket connection details.

**def start\_server():**

Starts a background thread server that writes keystrokes received to server\_log\_simple.txt.

**def send\_keystroke(key\_text):**

Sends each key pressed to the local server.

**def create\_session\_folder():**

Makes a timestamped folder for the session.

**def save\_session(folder\_name, keystrokes):**

Writes keystrokes to keystrokes.txt inside the session folder.

**def run\_single\_session():**

Captures keystrokes until Esc pressed, saves them at end.

**keyboard.Listener(on\_press=on\_press)**

Listens for keys; Esc stops session; all keys are logged.

**def main\_loop():**

Runs continuous loop of sessions, restarting after each ends, until Ctrl+C.

**if \_\_name\_\_ == '\_\_main\_\_': main\_loop()**

Program entry point.