Keylogger Code Documentation (Beginner-Friendly & Viva Ready)

#1. Introduction

This document explains a simple Python-based keystroke monitoring program using **pynput**.

It also includes viva-style questions, real-life analogies, and funny examples to make learning easier.

#2. What Does This Program Do?

This program:

- Records every key you press on the keyboard

- Saves those keys into a text file (`key_log.txt`)

- Shows the typed characters live on the screen

- Stops when you press the ESC key

Think of it like a **security camera for your keyboard**, capturing every movement.

Funny Example:

If your keyboard is gossiping about what you type, this program is basically the friend who writes everything in a diary.

#3. Libraries Used

##**pynput.keyboard**

This library lets Python listen to your keyboard events.

Real-life Example:

Imagine a friend quietly listening to every word you say.

That’s pynput, except it listens to your keyboard instead of your mouth.

##**datetime**

Used to add timestamps.

Real-life Example:

Just like CCTV footage shows exact times, this code records:

“Who typed what, and when?”

##**sys**

Used to print characters manually to the screen.

Funny Example:

Sys is like the waiter who brings your typed letters to the display table one by one.

#4. Important Variables

##`log_file = "key_log.txt"`

This is where all key logs are stored.

##`typed_text = []`

This list stores characters for managing the visual typing output.

Real-life Example:

Think of `typed_text` as your “undo history.”

#5. Function: on_press(key)

This function is called whenever a key is pressed.

##Steps inside:

1. **Get key name**

2. **Record timestamp**

3. **Write to log file**

4. **Display characters on screen**

5. **Special handling**

- Space

- Enter

- Backspace

Funny Example:

Backspace works like “Ctrl+Z in real life,” instantly erasing your mistakes.

#6. Function: on_release(key)

This function is triggered when a key is released.

When ESC is released → Program stops.

Real-life Example:

Like closing WhatsApp when you accidentally open your ex’s chat.

#7. Listener Loop

The last part creates a listener that keeps running until ESC is pressed.

#8. Viva Questions (Very Important)

##*Basic Questions*

1. What is a keylogger?

2. What is the use of pynput?

3. What is the role of the Listener class?

4. What happens when the ESC key is pressed?

##*Medium Questions*

5. Explain how timestamps are added.

6. Why do we handle backspace manually?

7. What is the purpose of `sys.stdout.write()`?

##*Advanced Questions*

8. Is this program harmful? How can it be misused?

9. What are ethical guidelines for using keyloggers?

10. How would you prevent illegal keylogging in real life?

#9. Concepts Used (Explained Simply)

##**Event Listeners**

These wait for an action (key press) and respond.

Real-life Example:

A dog waiting for you to say “walk” — then it reacts instantly.

##**File Handling**

Code writes each key into a file.

Real-life Example:

Like writing your daily diary, except the program does it automatically.

##**Conditional Logic**

Used to check which key was pressed.

Funny Example:

“If key == Space: insert awkward silence.”

#10. Summary

You now understand how the program:

- Listens to keys

- Logs them to a file

- Shows them on screen

- Stops when ESC is pressed

This project is perfect as a **beginner cybersecurity demonstration**, focusing on monitoring and logging—not misuse.