



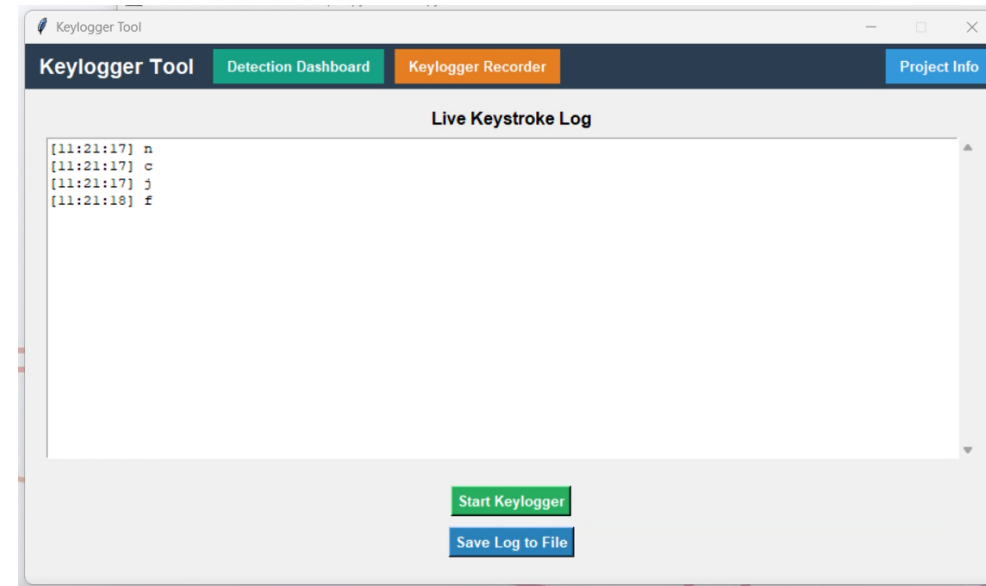
# KEYLOGGER

Internship Project by

Name	Employee ID	Email
J. Kavya sri	ST#IS#7459	kavyasrijangapalli@gmail.com
N. Varsha Reddy	ST#IS#7472	n.varshareddy070605@gmail.com
G. Rohith	ST#IS#7442	gangapuramrohith123@gmail.com
R. Keerthy	ST#IS#7476	routhukeerthy@gmail.com

# Introduction

- Keyloggers are malicious programs that capture keystrokes.
- They are often used for stealing sensitive information like passwords and credit card details.
- This project aims to detect suspicious processes and record keystrokes for security testing.
- Built using Python, Tkinter (GUI), and Psutil (process management).



# Objectives

- **Develop a desktop application that:**
  1. Scans processes for suspicious activity.
  2. Logs keystrokes in a safe test environment.
  3. Allows process termination from the dashboard.
  4. Displays live keystroke logs in a GUI.

# Technologies Used

- **Programming Language:** Python 3.x
- **Libraries:**
  - tkinter → GUI design
  - psutil → Process scanning & management
  - pynput → Keyboard event logging
  - threading → Background tasks
  - webbrowser, subprocess, shutil → File & PDF handling
  - **OS:** Windows
- **Tools:** Visual Studio Code / Python IDLE

# System Architecture

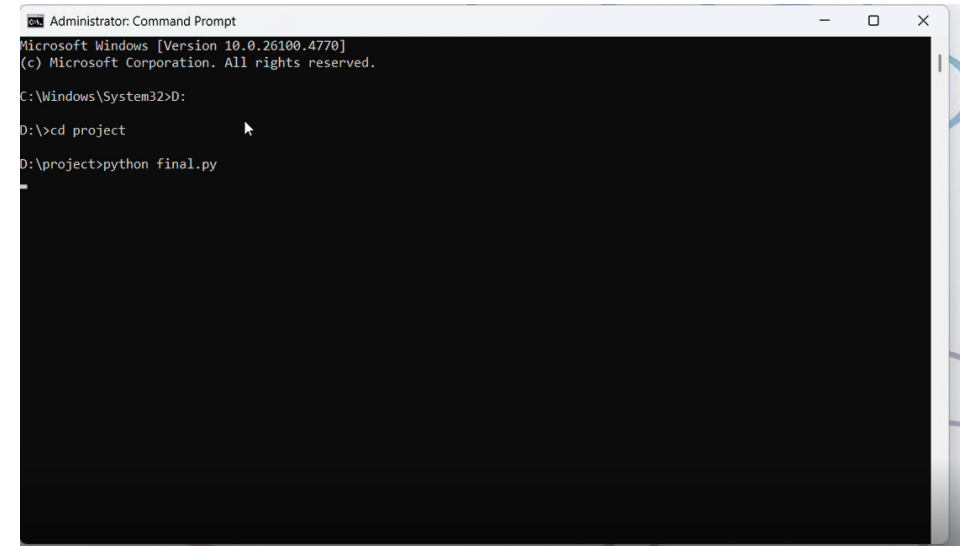
User Launches App	GUI loads
Dashboard Mode	Scans processes using psutil
Detection	Matches process names with suspicious keywords
Recorder Mode	Logs keystrokes using pynput listener
Output	Displays logs in real time + saves to file

# Implementation

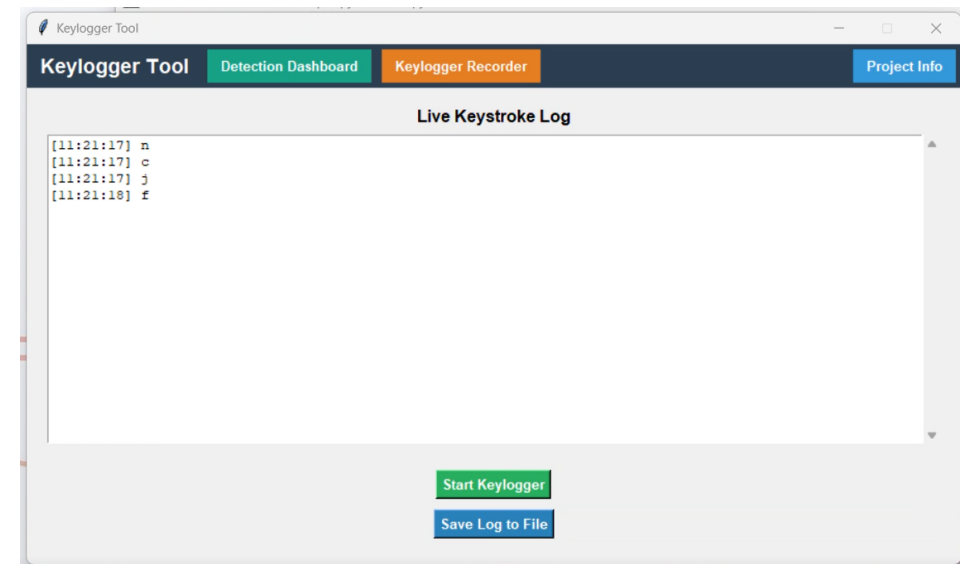
- **Dashboard Page:**
  - Lists suspicious processes with PID, Name, Executable, Command line.
  - Option to kill a selected process.
- **Recorder Page:**
  - Live keystroke display with timestamp.
  - Save logs to file.
- **Project Info Button:**
  - Opens PDF file with project documentation.
- **Admin Rights Check:**
  - Alerts user if not running as administrator.

# Results & Demo

- Successfully detected simulated suspicious processes.
- Real-time keylogging in test environment works as expected.
- Keystroke logs saved for later analysis.
- User-friendly interface with separate detection and recording modules.

A screenshot of a Windows Command Prompt window titled "Administrator: Command Prompt". The window shows the following text:

```
Microsoft Windows [Version 10.0.26100.4770]  
(c) Microsoft Corporation. All rights reserved.  
  
C:\Windows\System32>D:  
D:\>cd project  
D:\project>python final.py
```



# Challenges Faced

- **Module Installation Issues** – Different Python environments caused `ModuleNotFoundError`.
- **Admin Rights Requirement** – Some process details were inaccessible without administrator privileges.
- **Antivirus Interference** – Keylogger functionality triggered security warnings.
- **Cross-Platform Limitations** – Script primarily tested on Windows, limited support for other OS.



# Applications

- **Educational Tool** – Teaching students about malware behavior and detection methods.
- **Cybersecurity Training** – Simulating threats in a safe environment for practical learning.
- **System Monitoring** – Tracking suspicious processes for personal system security.
- **Research & Development** – Foundation for more advanced malware detection tools.

# Conclusion

1. Successfully developed a Python-based tool for detecting suspicious processes and simulating keylogger behavior in a controlled environment.
2. The application provides a user-friendly interface using Tkinter, making it accessible for non-technical users.
3. Real-time process scanning and keystroke logging features demonstrate practical cybersecurity concepts.
4. The project highlights the importance of awareness and proactive detection against hidden keyloggers.
5. Serves as an educational platform for students and researchers to study keylogger detection techniques safely.

THANK YOU