Practical No. 3B

Logging using Python

**Q.1]** Write a Python program to create a batch file for logging the ongoing processes in Windows.

**Steps:**

1. Run the Python code
2. Run the batch file as Admin
3. Open the created html file
4. (check paths in code carefully and the cmd command)

**Code:**

import os

try:

cmd = "WMIC /OUTPUT:\\HTML\\process\_log.html PROCESS get name,processid,creationdate /FORMAT:hform"

filepath = "C:\\Users\\Vaishanvi\\Desktop\\Siddhesh\\SEM V\\SEM\_V\\Codes\\EH\\P3\\Logs\\mycmd\_py.bat"

with open(filepath, 'w') as f:

f.write(cmd)

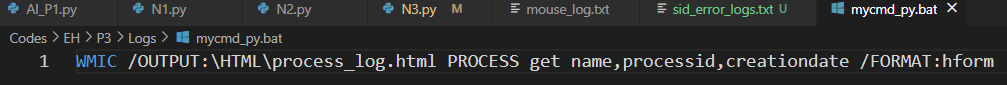
os.system("start "+filepath)

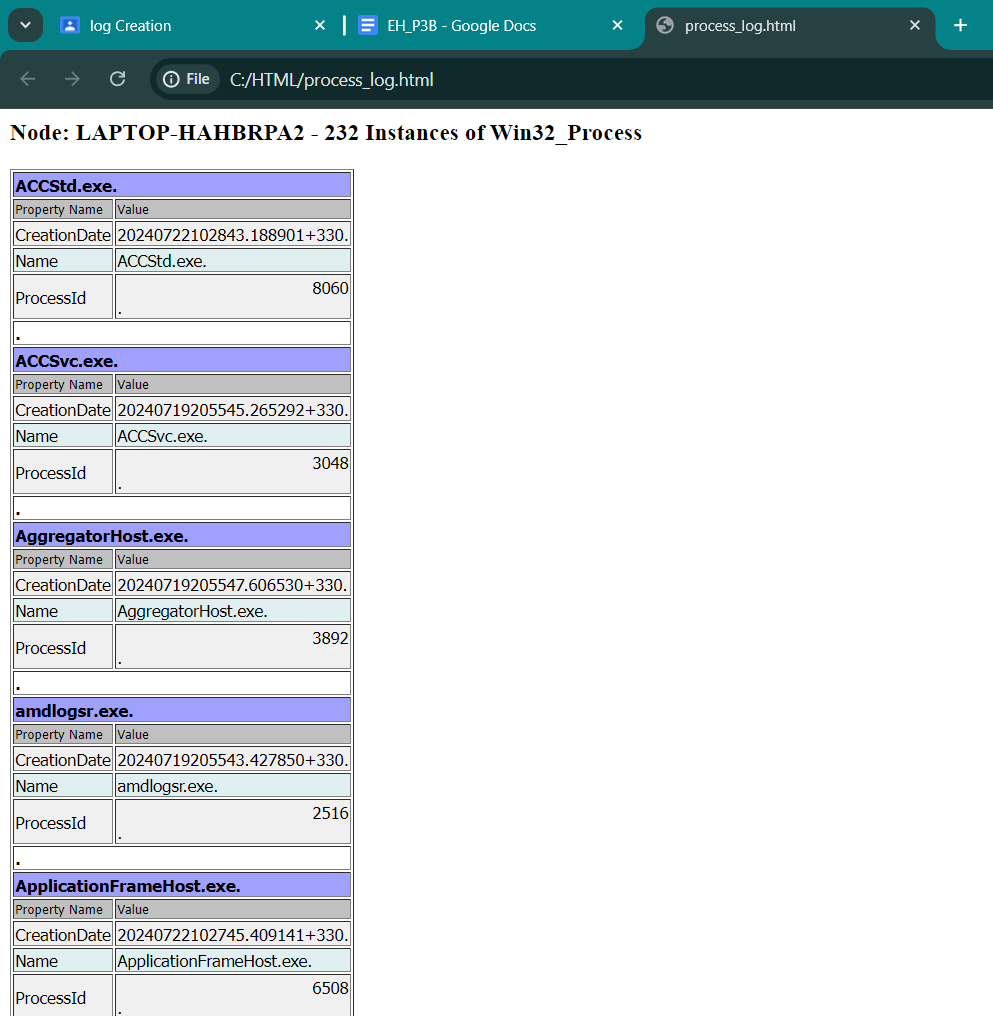
print("File created. Run log.")

except Exception as err:

print(err)

**Output:**

****



**Q.2]** Write a Python program to log mouse actions and store them in a .txt file.

**Code:**

from pynput.mouse import Listener

import logging

logging.basicConfig(filename="C:\\Users\\Vaishanvi\\Desktop\\Siddhesh\\SEM V\\SEM\_V\\Codes\\EH\\P3\\Logs\\mouse\_log.txt", level=logging.DEBUG, format='%(asctime)s:%(message)s')

def mouse\_move(x,y):

logging.info("Mouse moved to: ({0},{1})".format(x,y))

def mouse\_click(x,y,button,pressed):

if pressed:

logging.info("Mouse clicked at ({0}{1}) at {2}".format(x,y,button))

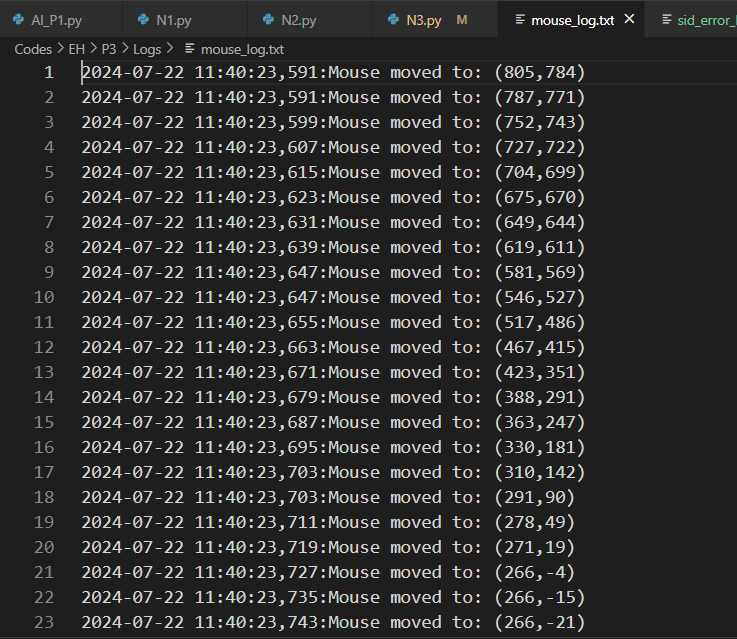
def mouse\_scroll(x,y,dx,dy):

logging.info("Mouse scorlled from ({0},{1}) to ({2}{3})".format(x,y,dx,dy))

with Listener(on\_move=mouse\_move, on\_click=mouse\_click, on\_scroll=mouse\_scroll) as listener:

listener.join()

**Output:**



**Q.3]** Write a Python program to log errors in the program in .txt file.

**Code:**

import logging

logging.basicConfig(filename="C:\\Users\\Vaishanvi\\Desktop\\Siddhesh\\SEM V\\SEM\_V\\Codes\\EH\\P3\\Logs\\sid\_error\_logs.txt",

level=logging.DEBUG,format='%(asctime)s %(levelname)s %(name)s %(message)s')

logger = logging.getLogger(\_\_name\_\_)

try:

1/0

except ZeroDivisionError as err:

logger.error(err)

try:

a,b = "hello",10

c = a + b

except TypeError as err:

logger.error(err)

try:

llist = [1, 3]

print(llist[5])

except IndexError as err:

logger.error(err)

try:

nn = "hello"

nn.reverse()

except AttributeError as err:

logger.error(err)

try:

x = (1,2,3)

x[0] = 8

except Exception as err:

logger.error(err)

**Output:**

