In [1]:
```
report = open("report.txt", "w")
report.write("Mindfull Monitor Report\n\n\n")
report.close()
```

In [2]:
```
import numpy as np
import cv2
import pyautogui
import time
import matplotlib.pyplot as plt
import io
import os
from google.cloud import vision
from PIL import Image
from pytesseract import pytesseract
from better_profanity import profanity
import tkinter, win32api, win32con, pywintypes
import datetime
```

In [3]:
```
os.environ['GOOGLE_APPLICATION_CREDENTIALS'] = "C:/Users/EEDA/PycharmProjects/
path_to_tesseract = r'C:\Program Files\Tesseract-OCR\tesseract.exe'
```

In [4]:
```
client = vision.ImageAnnotatorClient()
pytesseract.tesseract_cmd = path_to_tesseract
root = tkinter.Tk()
```

In [5]:
```
def update_label(txt):
    label.config(text=txt)
    label.after(10, update_label)
```

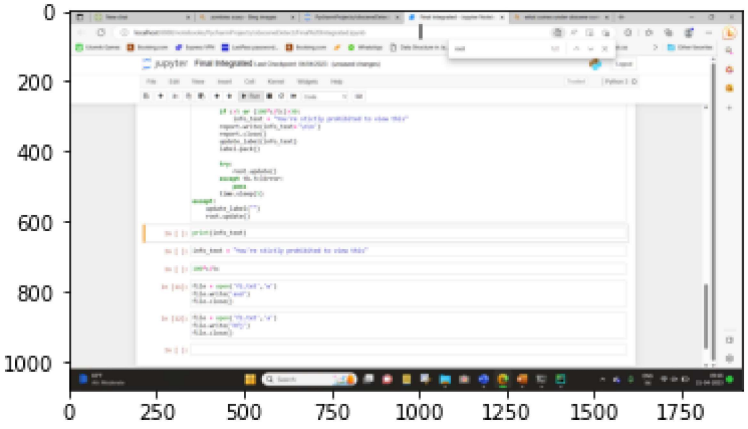
In [6]:
```
label = tkinter.Label(text='You are under safe monitoring', font=('Times New R
label.master.overrideredirect(True)
label.master.geometry("+1000+40")
label.master.lift()
label.master.wm_attributes("-topmost", True)
label.master.wm_attributes("-disabled", True)
label.master.wm_attributes("-transparentcolor", "white")
label.bind("<Escape>", lambda e:sys.exit())
hWindow = pywintypes.HANDLE(int(label.master.frame(), 16))

exStyle = win32con.WS_EX_COMPOSITED | win32con.WS_EX_LAYERED | win32con.WS_EX_
win32api.SetWindowLong(hWindow, win32con.GWL_EXSTYLE, exStyle)
```

Out[6]: 524296

In [ ]:

```
try:
    while True:
        report = open('report.txt','a')
        now = datetime.datetime.now()
        report.write('Session: '+str(now)+'\n')


        image = pyautogui.screenshot()
        image = cv2.cvtColor(np.array(image),cv2.COLOR_RGB2BGR)
        cv2.imwrite("captured.png", image)
        plt.imshow(image)
        plt.show()

        file_name = os.path.abspath('captured.png')
        with io.open(file_name, 'rb') as image_file:
            content = image_file.read()
        image = vision.Image(content=content)
        response = client.safe_search_detection(image=image)

        safe_limits=[]
        safe_limits.append(str(response.safe_search_annotation.adult))
        safe_limits.append(str(response.safe_search_annotation.spoof))
        safe_limits.append(str(response.safe_search_annotation.medical))
        safe_limits.append(str(response.safe_search_annotation.violence))
        safe_limits.append(str(response.safe_search_annotation.racy))
        print(safe_limits,end='\n')

        if  'Likelihood.VERY_LIKELY' in safe_limits or 'Likelihood.LIKELY' in
            info_text = "You're stictly prohibited to view this"
        elif 'Likelihood.UNKNOWN' in safe_limits or 'Likelihood.POSSIBLE' in s
            info_text = "You're at risk of viewing obscene content"
        elif 'Likelihood.VERY_UNLIKELY' in safe_limits or 'Likelihood.UNLIKELY
            info_text = "Image content is safe"

        img = Image.open('captured.png')
        text = pytesseract.image_to_string(img)

        tc=0
        c=0
        for i in text.split(" "):
            if profanity.contains_profanity(i):
                c=c+1
            tc=tc+1
        print(c,"words contains profanity of",tc)

        if c>5 or (100*c/tc)>30:
            info_text = "You're stictly prohibited to view this"
        report.write(info_text+'\n\n')
        report.close()
        update_label(info_text)
        label.pack()

        try:
            root.update()
        except tk.TclError:
            pass
        time.sleep(5)
```

```
except:
    update_label("")
    root.update()
```




```
['Likelihood.VERY_UNLIKELY', 'Likelihood.VERY_UNLIKELY', 'Likelihood.VERY_UNL
IKELY', 'Likelihood.VERY_UNLIKELY', 'Likelihood.VERY_UNLIKELY']
1 words contains profanity of 153
```

In [ ]:
```
print(info_text)
```

In [ ]:
```
info_text = "You're stictly prohibited to view this"
```

In [ ]:
```
100*c/tc
```

In [11]:
```
file = open('f1.txt','w')
file.write('asd')
file.close()
```

In [12]:
```
file = open('f1.txt','a')
file.write('hfj')
file.close()
```

In [ ]:
```

```