

EXPERIMENT : 2
REG NO 21222230003
NAME: ABISHEK PV

```
from IPython.display import display, Javascript
from google.colab.output import eval_js
from base64 import b64decode
import cv2
import numpy as np
from google.colab.patches import cv2_imshow
```

```
def capture_frame():
    js = Javascript('''
    async function takePhoto() {
        const div = document.createElement('div');
        const video = document.createElement('video');
        div.appendChild(video);
        document.body.appendChild(div);
        const stream = await navigator.mediaDevices.getUserMedia({video: true});
        video.srcObject = stream;
        await video.play();
        const canvas = document.createElement('canvas');
        canvas.width = video.videoWidth;
        canvas.height = video.videoHeight;
        canvas.getContext('2d').drawImage(video, 0, 0);
        stream.getTracks().forEach(track => track.stop());
        div.remove();
        return canvas.toDataURL('image/jpeg', 1.0);
    }
    ''')
    display(js)
    data = eval_js('takePhoto()')
    binary = b64decode(data.split(',')[1])
    nparr = np.frombuffer(binary, np.uint8)
    frame = cv2.imdecode(nparr, cv2.IMREAD_COLOR)
    return frame
```

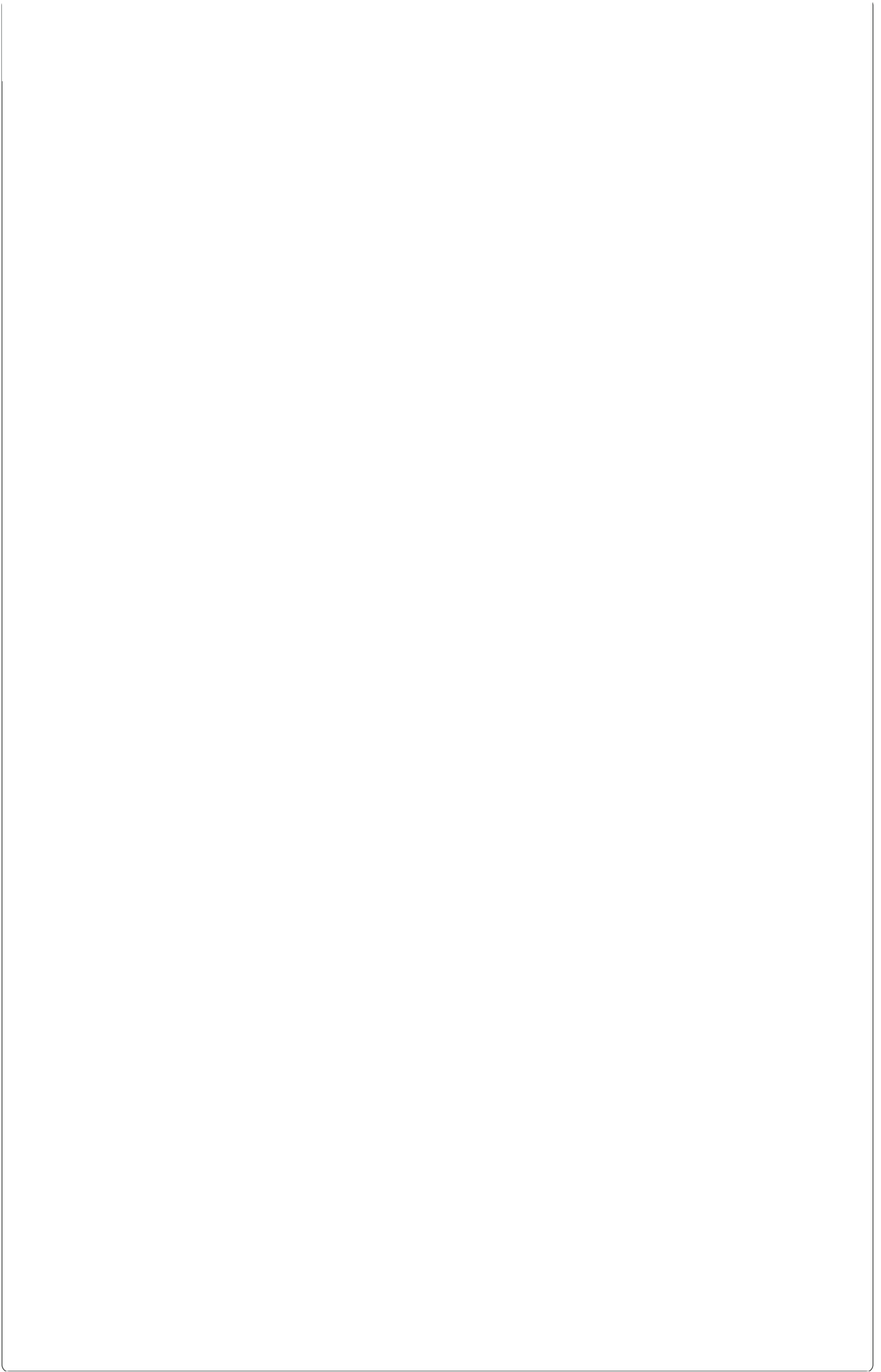
```
## i) Write the frame as JPG file
frame = capture_frame()
cv2.imwrite("captured_image.jpg", frame)
print("Image saved as captured_image.jpg")
cv2_imshow(frame)
```

```
## ii) Display the video
import time
#since Google Colab does not support continuous webcam streaming, a single captured frame is displayed to simulate video output
print("Starting pseudo video capture... (Press stop button to end cell)")
```

```
for i in range(5): # capture 5 frames as demo
    frame = capture_frame()
    cv2_imshow(frame)
    time.sleep(1) # wait 1 second before next frame
```

```
## iii) Display the video by resizing the window
frame = capture_frame()
height, width, _ = frame.shape
smaller_frame = cv2.resize(frame, (0,0), fx=0.5, fy=0.5)
image = np.zeros_like(frame)
image[:height//2, :width//2] = smaller_frame
cv2_imshow(image)
```

```
## iv) Rotate and display the video
frame = capture_frame()
rotated = cv2.rotate(frame, cv2.ROTATE_90_CLOCKWISE)
cv2_imshow(rotated)
```



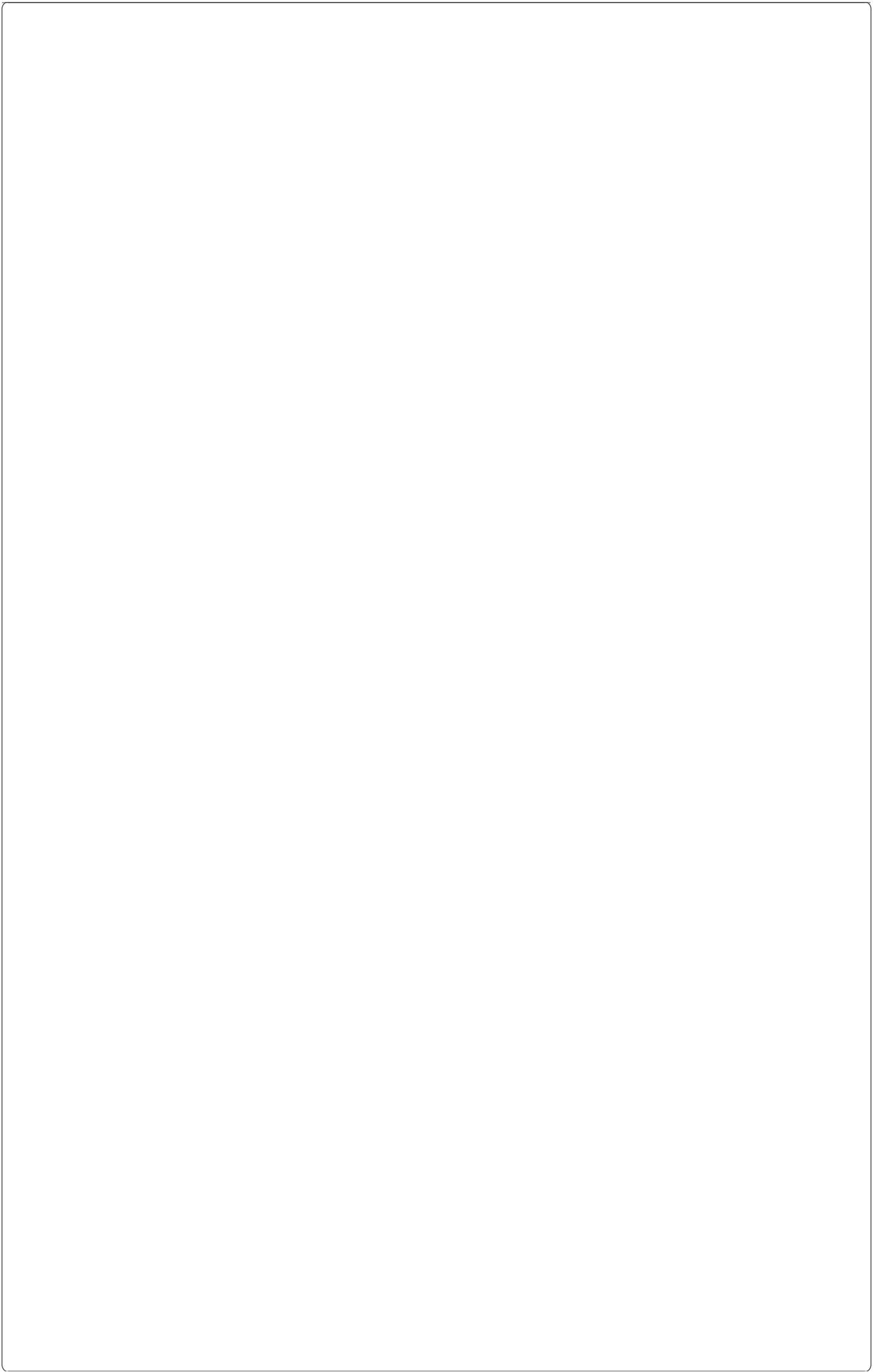


Image saved as captured_image.jpg



Starting pseudo video capture... (Press stop button to end cell)

