#first find the ip address that the raspberry pi is on

#type ifconfig in the terminal and find the inet addr

import io

import picamera

import logging

import socketserver

from threading import Condition

from http import server

PAGE="""\

<html>

<head>

<title>Raspberry Pi - Surveillance Camera</title>

</head>

<body>

<center><h1>Raspberry Pi - Surveillance Camera</h1></center>

<center><img src="stream.mjpg" width="640" height="480"></center>

</body>

</html>

"""

class StreamingOutput(object):

def \_\_init\_\_(self):

self.frame = None

self.buffer = io.BytesIO()

self.condition = Condition()

def write(self, buf):

if buf.startswith(b'\xff\xd8'):

# New frame, copy the existing buffer's content and notify all clients it's available

self.buffer.truncate()

with self.condition:

self.frame = self.buffer.getvalue()

self.condition.notify\_all()

self.buffer.seek(0)

return self.buffer.write(buf)

class StreamingHandler(server.BaseHTTPRequestHandler):

def do\_GET(self):

if self.path == '/':

self.send\_response(301)

self.send\_header('Location', '/index.html')

self.end\_headers()

elif self.path == '/index.html':

content = PAGE.encode('utf-8')

self.send\_response(200)

self.send\_header('Content-Type', 'text/html')

self.send\_header('Content-Length', len(content))

self.end\_headers()

self.wfile.write(content)

elif self.path == '/stream.mjpg':

self.send\_response(200)

self.send\_header('Age', 0)

self.send\_header('Cache-Control', 'no-cache, private')

self.send\_header('Pragma', 'no-cache')

self.send\_header('Content-Type', 'multipart/x-mixed-replace; boundary=FRAME')

self.end\_headers()

try:

while True:

with output.condition:

output.condition.wait()

frame = output.frame

self.wfile.write(b'--FRAME\r\n')

self.send\_header('Content-Type', 'image/jpeg')

self.send\_header('Content-Length', len(frame))

self.end\_headers()

self.wfile.write(frame)

self.wfile.write(b'\r\n')

except Exception as e:

logging.warning(

'Removed streaming client %s: %s',

self.client\_address, str(e))

else:

self.send\_error(404)

self.end\_headers()

class StreamingServer(socketserver.ThreadingMixIn, server.HTTPServer):

allow\_reuse\_address = True

daemon\_threads = True

with picamera.PiCamera(resolution='640x480', framerate=24) as camera:

output = StreamingOutput()

#Uncomment the next line to change your Pi's Camera rotation (in degrees)

#camera.rotation = 90

camera.start\_recording(output, format='mjpeg')

try:

address = ('', 8000)

server = StreamingServer(address, StreamingHandler)

server.serve\_forever()

finally:

camera.stop\_recording()