Logo, company name

Description automatically generated

Bangabandhu Sheikh Mujibur Rahman Digital University, Bangladesh

Faculty of Cyber Physical System

Dept. of Internet of Things and Robotics Engineering (IRE)

Course Title: Real life in IoT Lab

Course Code: IOT 4316

**Lab Report-04**

**Submitted to-**

**Teacher name:** Nurjahan Nipa

**Designation:** Lecturer

**Department:** IRE

**Submitted by-**

Md. Shahriar Hossain Apu (1901036)

Date of Submission: 03-08-2023

**Task:**Raspberry pi setup and pi camera interfacing,image capturing,video recording and live sreaming.

**Introduction:**

Raspberry Pi is a series of small, single-board computers developed in the United Kingdom by the Raspberry Pi Foundation. It was initially created to promote computer science education and to provide an affordable platform for learning programming and tinkering with electronics. Since its inception, Raspberry Pi has gained immense popularity in various fields due to its versatility, compact size, low cost, and extensive community support.

The Raspberry Pi Camera Module is an official camera accessory specifically designed for Raspberry Pi boards. It allows users to capture images and record videos with ease, making it an excellent tool for projects involving photography, surveillance, computer vision, and more.

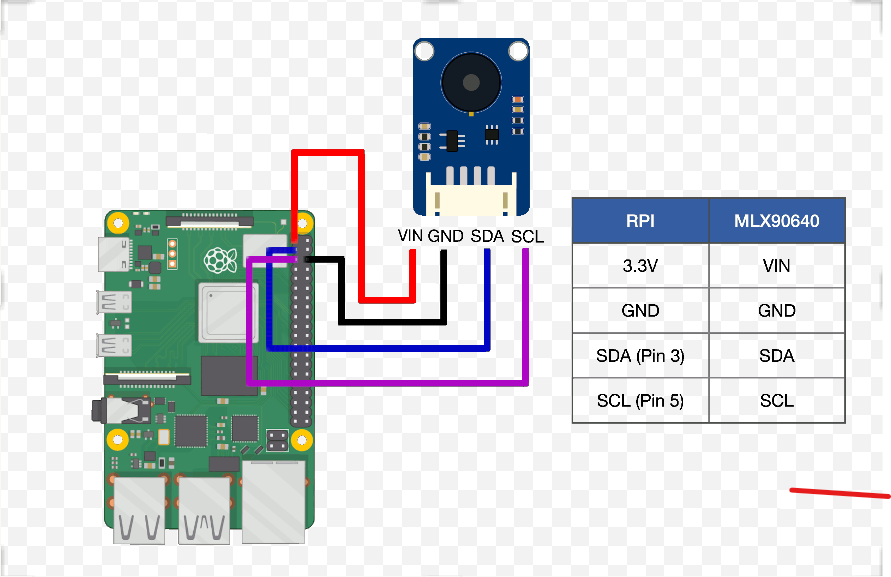
**Apparatus:**

1.Raspberry pi 3 model B+

2.Pi camera module

3.B type cable.

**Circuit Diagram:**

****

**Fig:**Circuit diagram of raspberry pi with camera module.

**Working Procedure:**

**Part 1: Raspberry Pi Setup**

Raspberry Pi is a credit card-sized single-board computer developed in the UK by the Raspberry Pi Foundation. It is a versatile device that can be used for a wide range of projects, from simple DIY gadgets to advanced robotics and IoT applications. Setting up a Raspberry Pi involves a few key steps:

**Get the Hardware:** Obtain a Raspberry Pi board, a microSD card (at least 8GB), a power supply, and optionally, a case and any additional peripherals like a keyboard, mouse, and monitor.

**Download the Operating System:** Raspberry Pi requires an operating system to function. Raspbian (now known as Raspberry Pi OS) is the official and most commonly used OS. Download the latest version of Raspberry Pi OS from the Raspberry Pi Foundation's website.

**Flash the OS onto the microSD Card:** Use a computer and an SD card writer to flash the downloaded OS image onto the microSD card. There are various software tools available for this purpose, such as Etcher.

**Boot Up the Raspberry Pi:** Insert the microSD card into the Raspberry Pi, connect the power supply, and the Raspberry Pi will boot up.We will see the Raspberry Pi OS desktop on your monitor.

**Configure the OS**: Follow the on-screen instructions to set our preferred language, time zone, password, and Wi-Fi network. This will complete the initial setup of your Raspberry Pi.

**Part 2: Pi Camera Interfacing**

The Raspberry Pi Camera Module allows us to capture images and record videos with your Raspberry Pi. Interfacing the Pi Camera involves the following steps:

**Get a Compatible Camera Module:** We have a Raspberry Pi Camera Module compatible with our Raspberry Pi model. There are two types of camera modules and we have the standard camera module.

**Enable the Camera Interface:** By default, the camera interface is disabled on the Raspberry Pi. We need to enable it through the Raspberry Pi Configuration tool or by running the raspi-config command in the terminal.

**Install Camera Software:** In most cases, the camera software is pre-installed with Raspberry Pi OS. However, we can ensure that it's up to date by running some following commands in the terminal.

**Capture Images and Videos:** Now that the camera interface is enabled, we can use various programming languages (Python is the most common) to interact with the camera and capture images or record videos. Python provides libraries like picamera to easily control the camera module.

**Sketch:**

**Live streaming**

import io

import picamera

import logging

import socketserver

from threading import Condition

from http import server

PAGE="""\

<html>

<head>

<title>Raspberry Pi - Live Vedio Streaming</title>

</head>

<body>

<center><h1>Raspberry Pi - live Vedio Streaming</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)

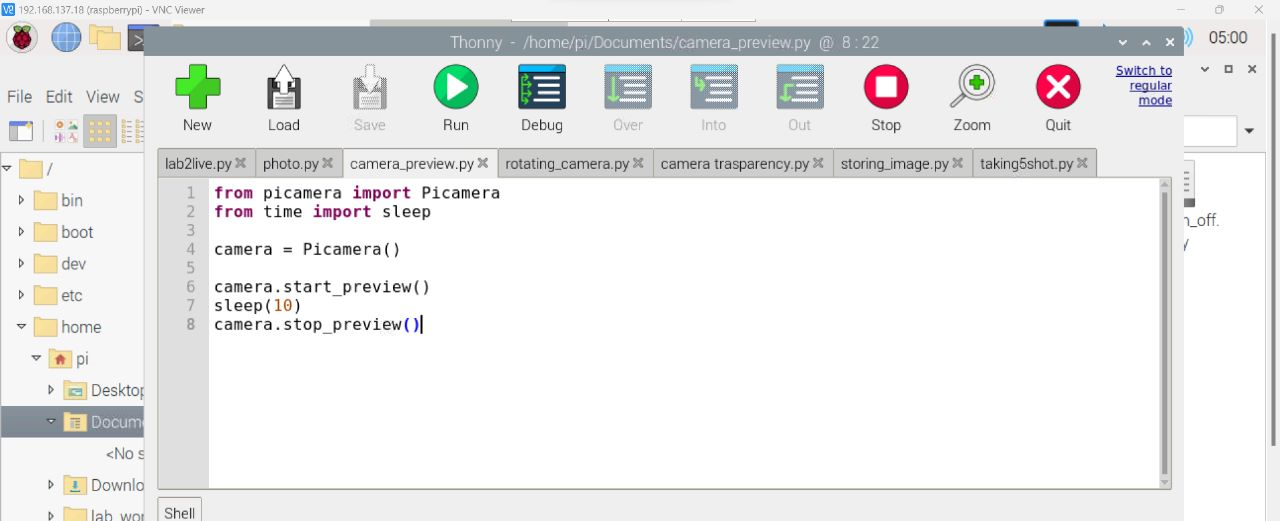
server = StreamingServer(address, StreamingHandler)

server.serve\_forever()

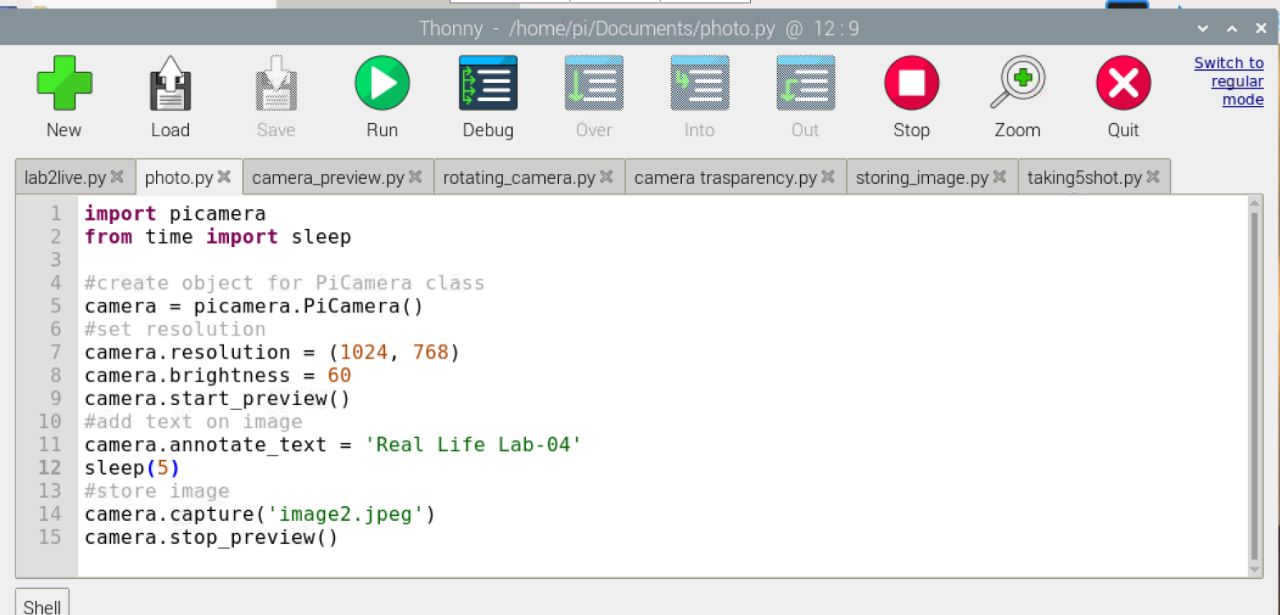
finally:

camera.stop\_recording()

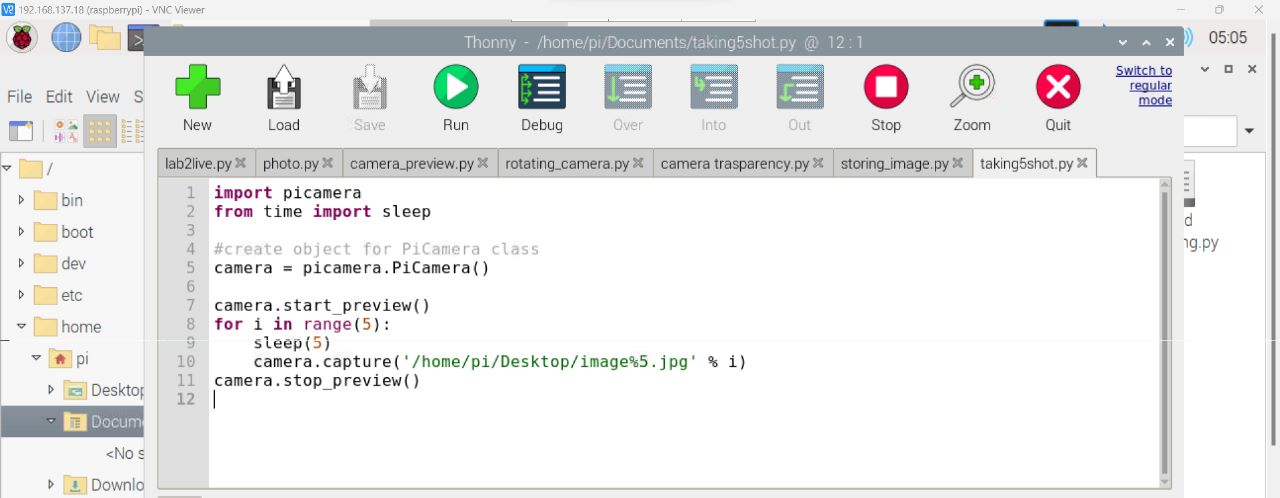
**Camera preview**



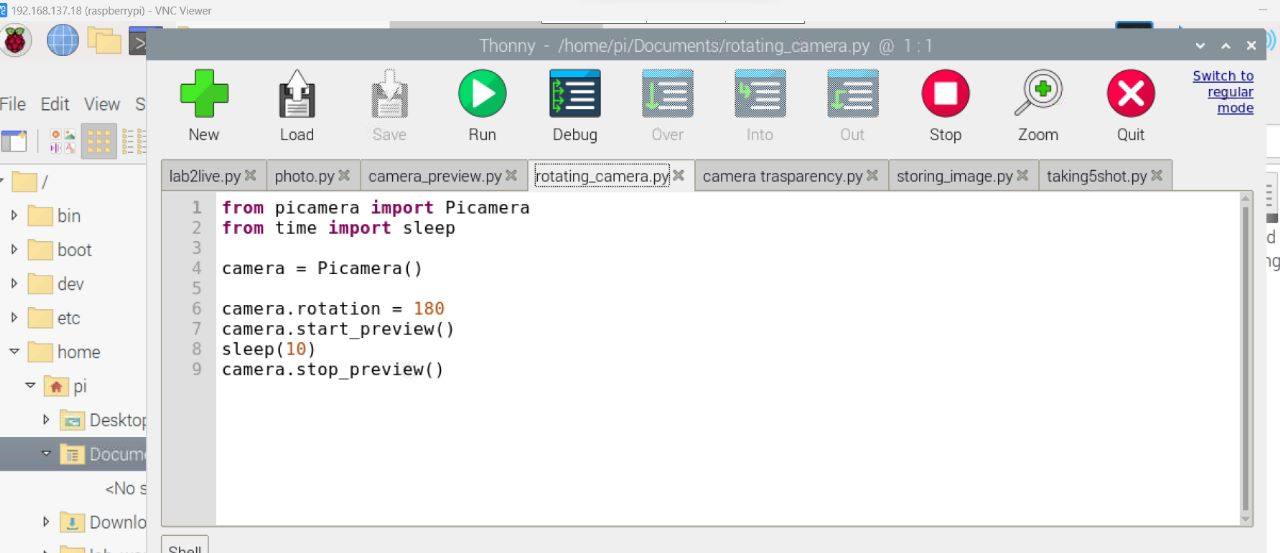
**Capturing photo**



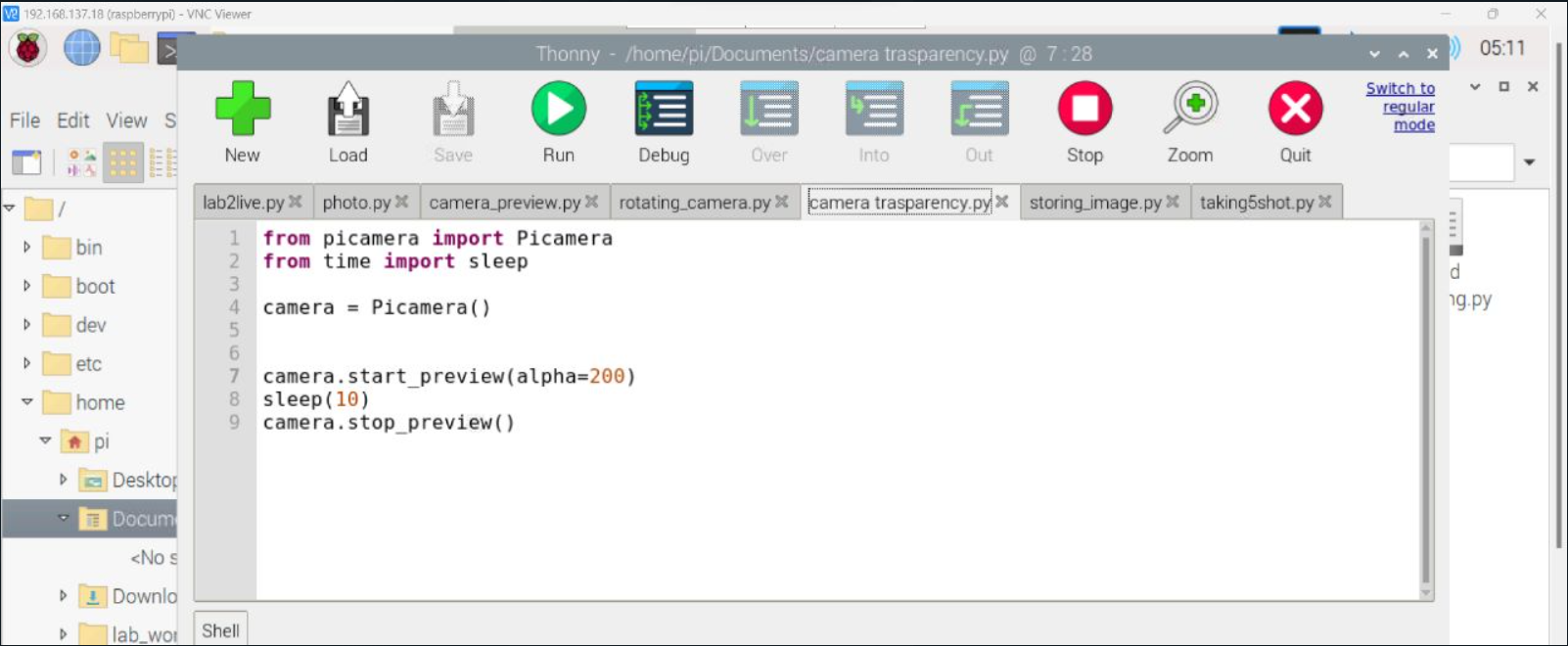
**Taking five shoot**



**Rotating camera**



**Camera tranasparency**

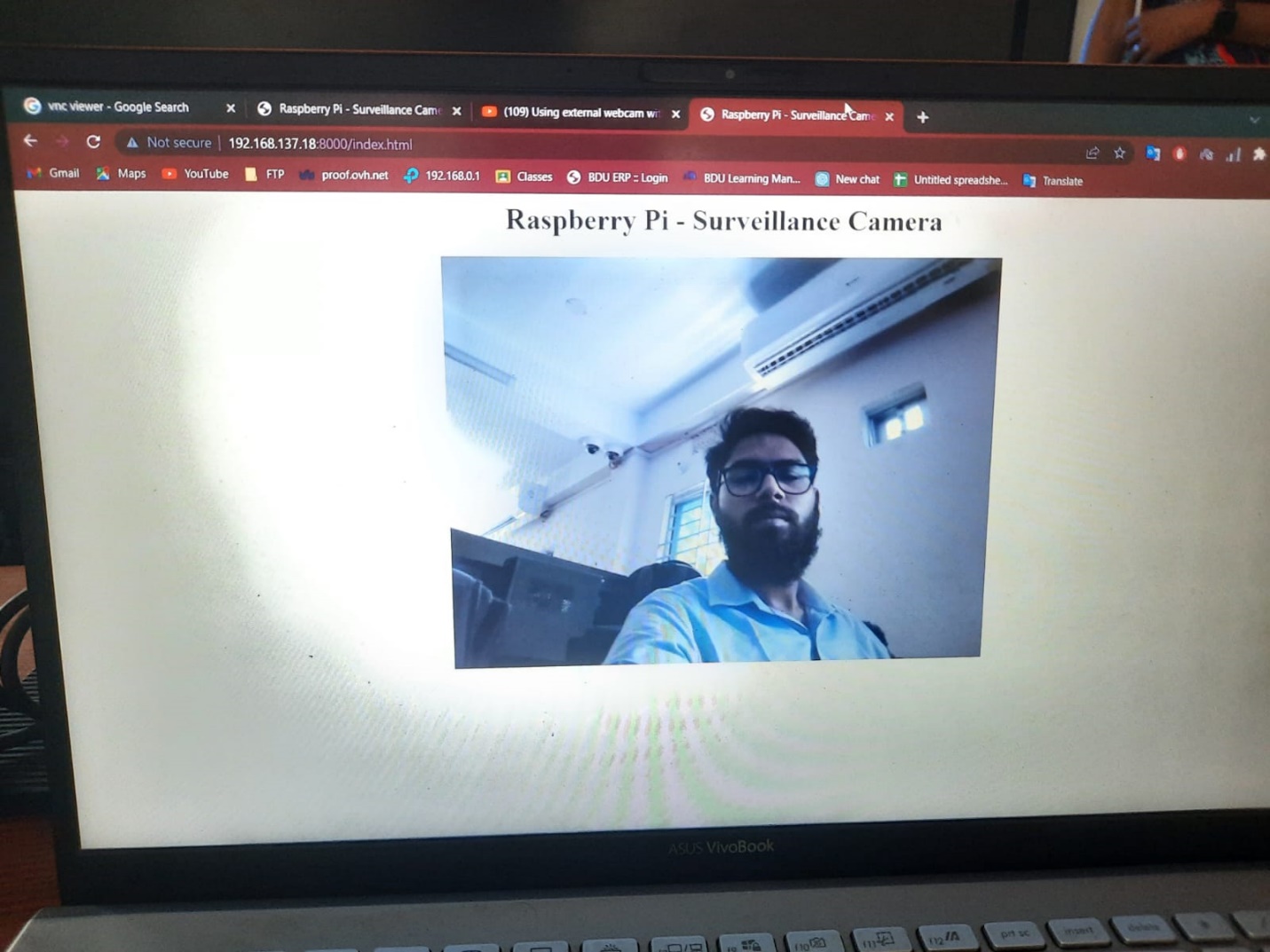
****

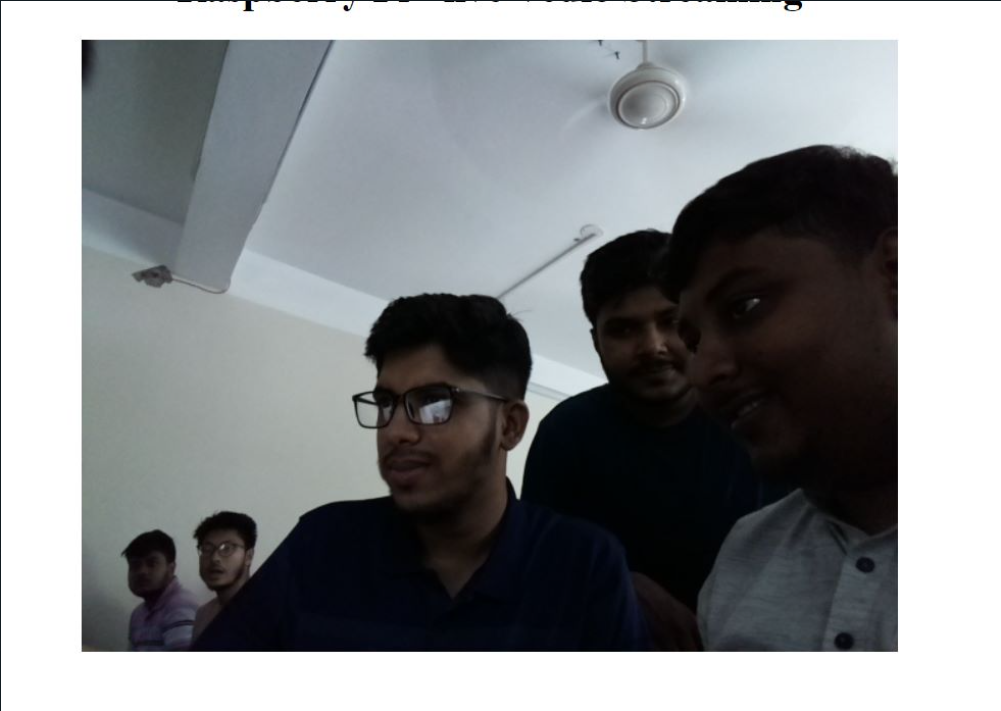
**Video link**

[**https://drive.google.com/file/d/1GgB9b0Ts2Sv2zZIebnlaoHggnNFBa1kD/view?usp=sharing**](https://drive.google.com/file/d/1GgB9b0Ts2Sv2zZIebnlaoHggnNFBa1kD/view?usp=sharing)

**Operational view:**

****





**Fig:** Operational view of capturing live image with raspberry pi camera module.

**Conclusion:**

In conclusion, capturing live images with the Raspberry Pi Camera Module opens exciting opportunities for creative projects and experimentation. Whether we are a beginner exploring the world of electronics or an experienced developer working on advanced applications, the combination of Raspberry Pi and its camera module empowers you to dive into the realm of digital imaging and unleash our creativity.

***–End of the Report–***