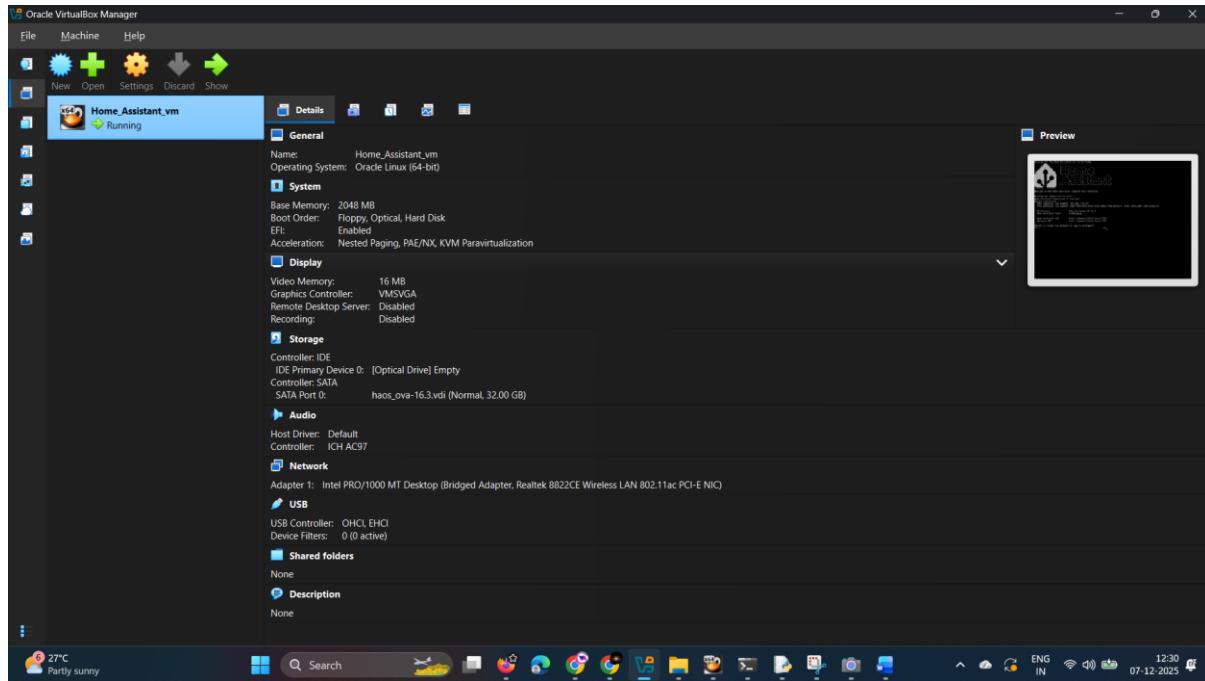


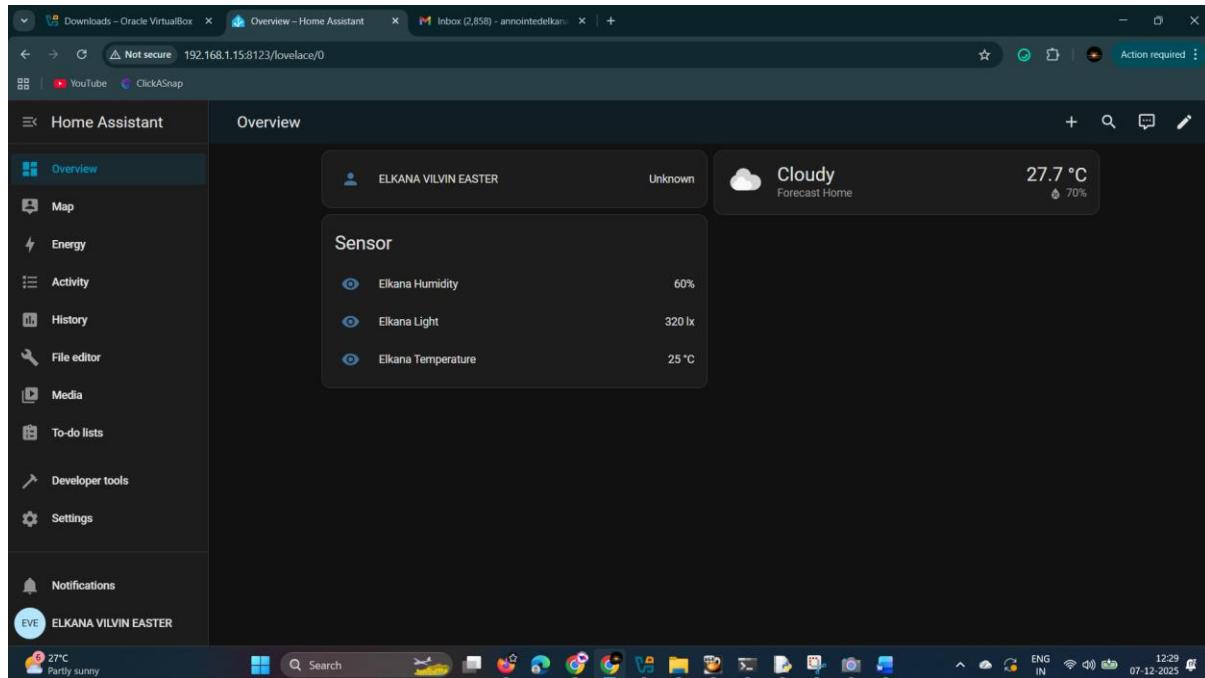
NAME: ELKANA VILVIN EASTER

Register Number: 42611034

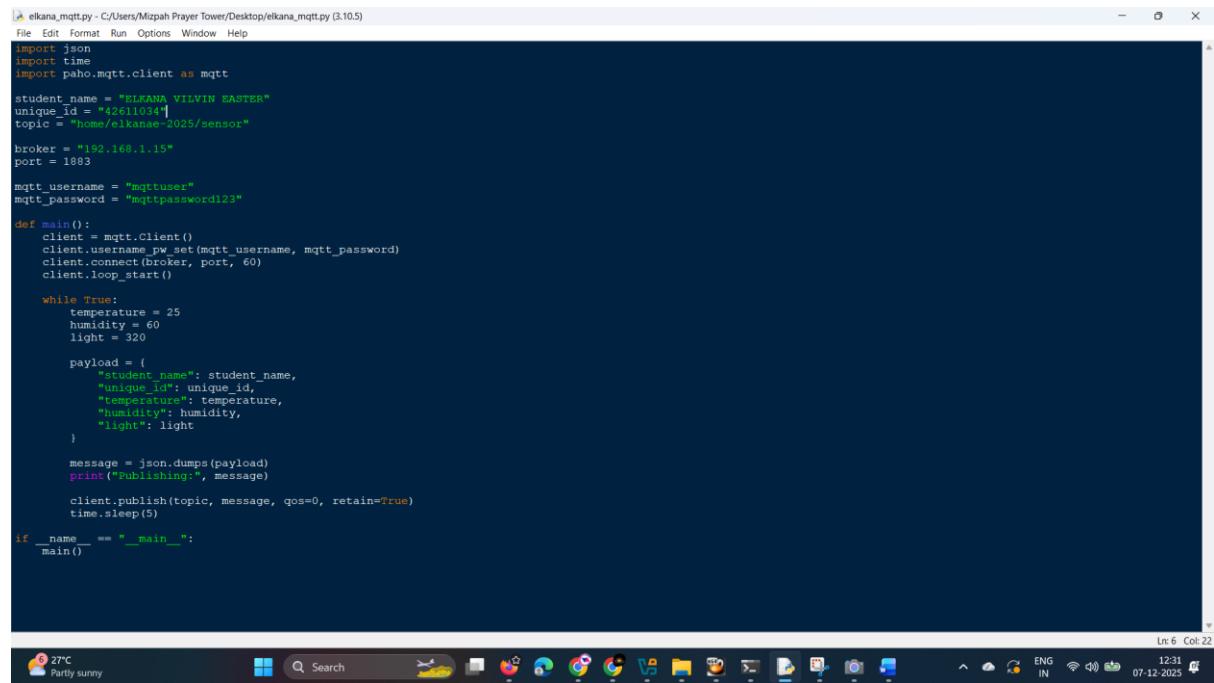
Virtual Box (Machine)



Dashboard



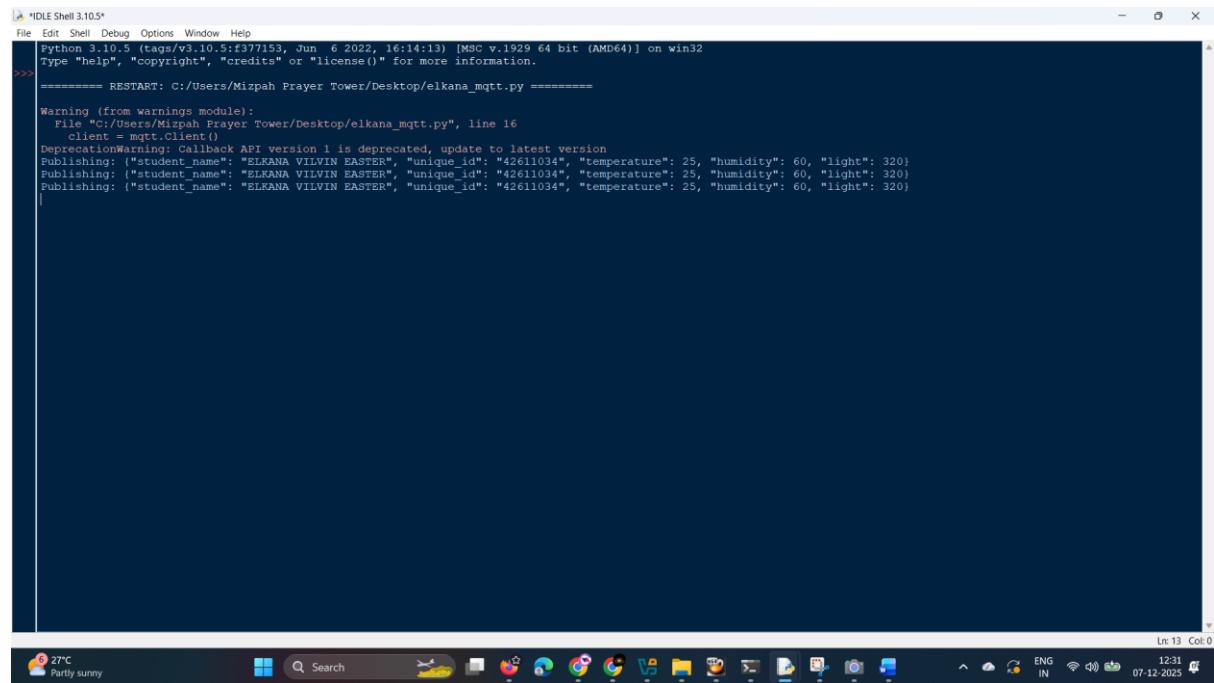
Python Script



```
elkana_mqtt.py - C:/Users/Mizpah Prayer Tower/Desktop/elkana_mqtt.py (3.10.5)
File Edit Format Run Options Window Help
import json
import time
import paho.mqtt.client as mqtt
student_name = "ELKANA VILVIN EASTER"
unique_id = "42611034"
topic = "home/elkanae-2025/sensor"
broker = "192.168.1.15"
port = 1883
mqtt_username = "mqttuser"
mqtt_password = "mqtpassword123"
def main():
    client = mqtt.Client()
    client.username_pw_set(mqtt_username, mqtt_password)
    client.connect(broker, port, 60)
    client.loop_start()
    while True:
        temperature = 25
        humidity = 60
        light = 320
        payload = {
            "student_name": student_name,
            "unique_id": unique_id,
            "temperature": temperature,
            "humidity": humidity,
            "light": light
        }
        message = json.dumps(payload)
        print("Publishing:", message)
        client.publish(topic, message, qos=0, retain=True)
        time.sleep(5)
if __name__ == "__main__":
    main()
```

The screenshot shows a Windows desktop environment. A terminal window titled 'elkana_mqtt.py' is open, displaying the Python script. The script connects to an MQTT broker at '192.168.1.15' port 1883, publishing sensor data (temperature, humidity, light) every 5 seconds. The desktop taskbar at the bottom shows various icons for applications like File Explorer, Google Chrome, and Microsoft Edge.

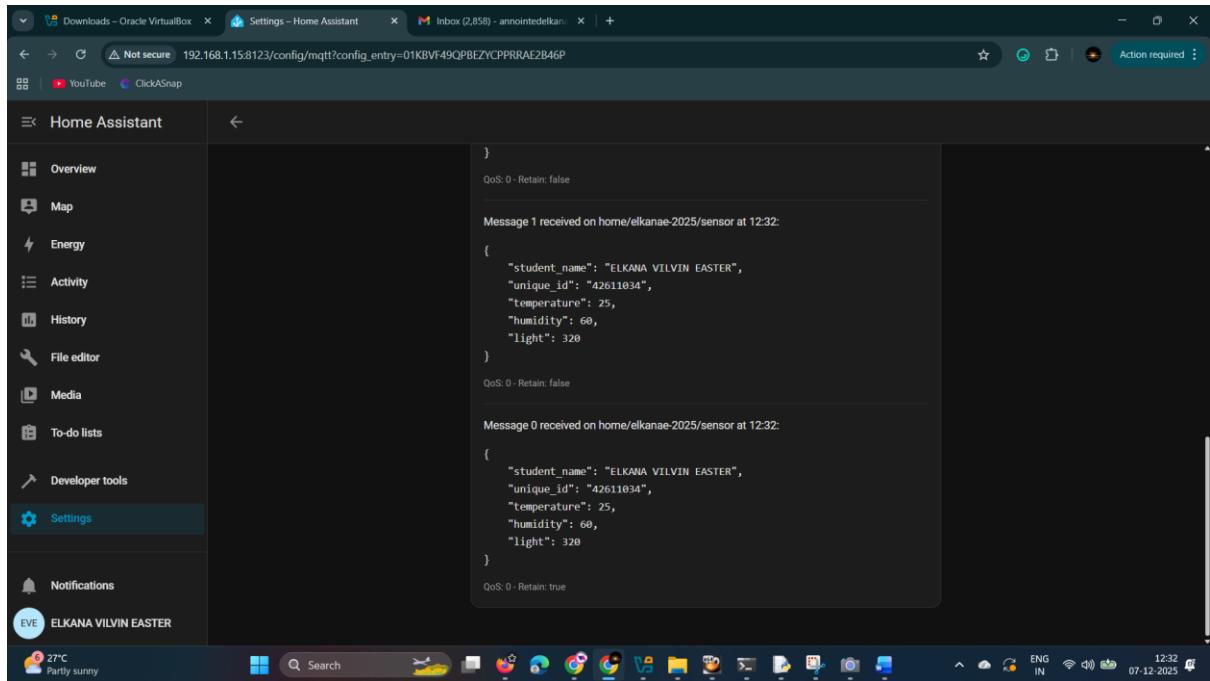
Output (Publishing)



```
IDLE Shell 3.10.5*
File Edit Shell Debug Options Window Help
Python 3.10.5 (tags/v3.10.5:f377153, Jun  6 2022, 16:14:13) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: C:/Users/Mizpah Prayer Tower/Desktop/elkana_mqtt.py =====
Warning (from warnings module):
  File "C:/Users/Mizpah Prayer Tower/Desktop/elkana_mqtt.py", line 16
    client = mqtt.Client()
DeprecationWarning: Calling API version 1 is deprecated, update to latest version
Publishing: {"student_name": "ELKANA VILVIN EASTER", "unique_id": "42611034", "temperature": 25, "humidity": 60, "light": 320}
Publishing: {"student_name": "ELKANA VILVIN EASTER", "unique_id": "42611034", "temperature": 25, "humidity": 60, "light": 320}
Publishing: {"student_name": "ELKANA VILVIN EASTER", "unique_id": "42611034", "temperature": 25, "humidity": 60, "light": 320}
```

The screenshot shows the same Windows desktop environment. An IDLE shell window is open, showing the execution of the 'elkana_mqtt.py' script. The output shows the script publishing sensor data (temperature, humidity, light) to the MQTT broker. The desktop taskbar at the bottom shows various icons for applications like File Explorer, Google Chrome, and Microsoft Edge.

MQTT Listening SS:



Extra sensor : Light sensor used



Explanation of this Project:

In this project, I installed Home Assistant inside a VirtualBox machine. After Home Assistant finished setting up, I installed the Mosquitto MQTT Broker add-on and enabled the MQTT integration. Then I wrote a Python script on my Windows laptop. This script sends temperature, humidity, and an extra sensor value (light) to a fixed MQTT topic. I added my name, register number, and topic in the script as required.

Home Assistant receives these MQTT messages through the Mosquitto broker. I created three MQTT sensors in the configuration.yaml file so Home Assistant can read the JSON values and show them on the dashboard.

After restarting Home Assistant, I could see all three values updating live on the dashboard while the Python script was running. This shows that MQTT communication between Python and Home Assistant is working correctly.

Conclusion Of this Project:

This assignment helped me understand how MQTT works and how Python, Home Assistant, and the MQTT broker connect together. I learned how to publish data using Python and how Home Assistant reads and displays it.

Everything worked successfully in the end — the values from my script updated live on the Home Assistant dashboard. This shows that my setup and configuration were correct.

This was a good learning experience and it helped me understand IoT communication in a simple, practical way.