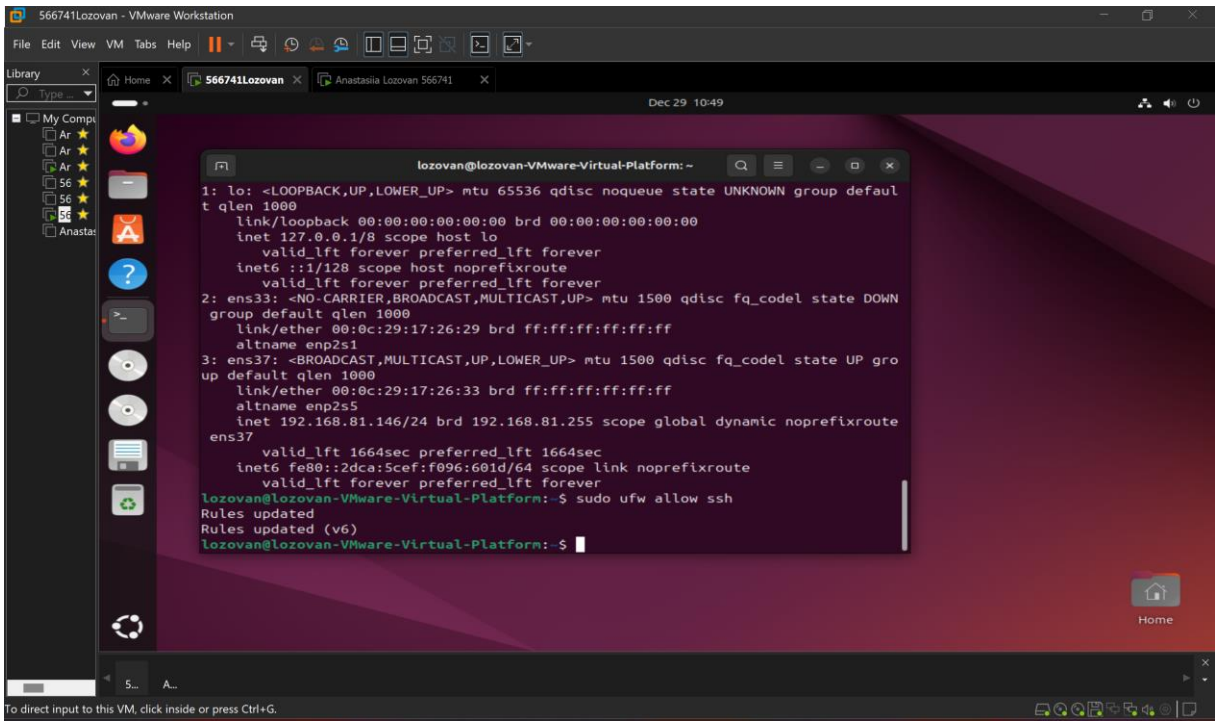
Template Week 6 – Networking

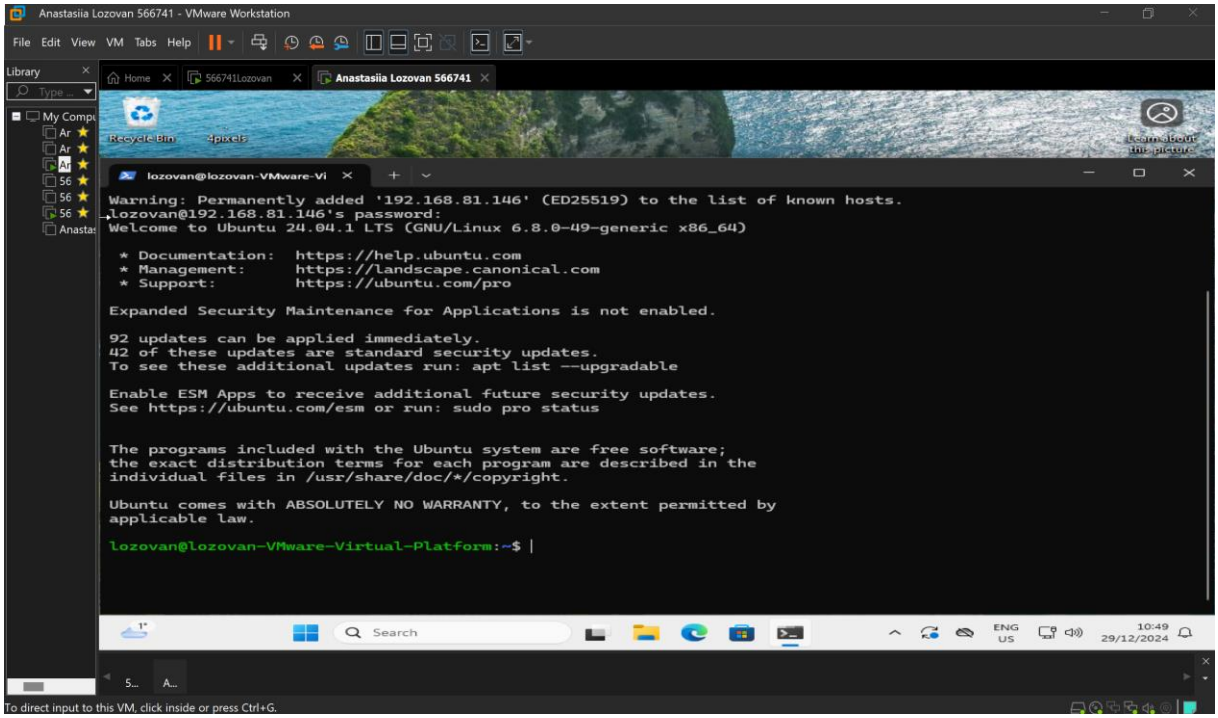
Student number:566741

Assignment 6.1: Working from home

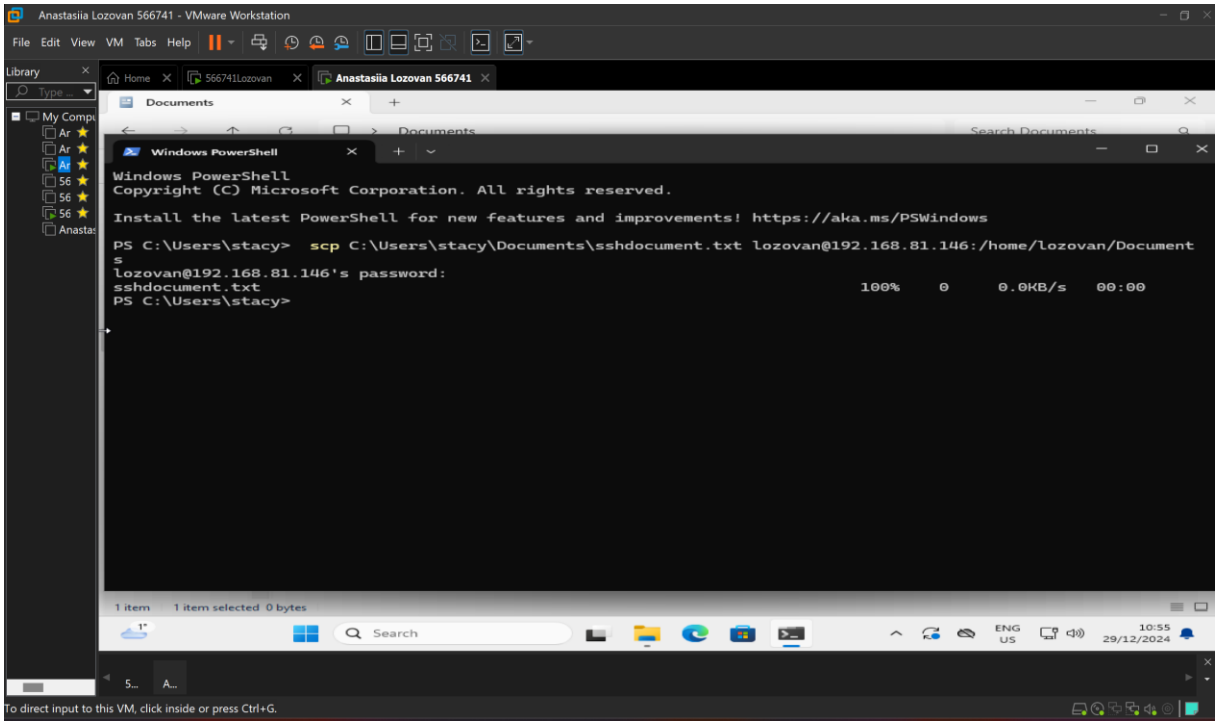
Screenshot installation openssh-server:

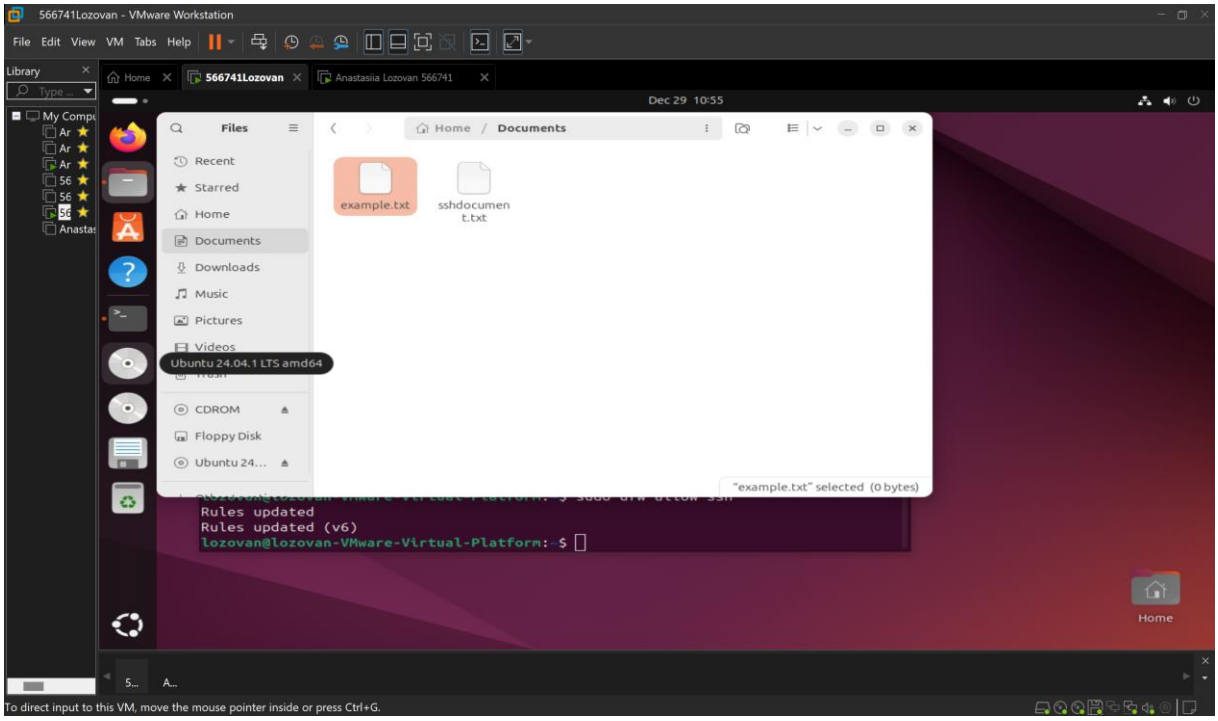


Screenshot successful SSH command execution:

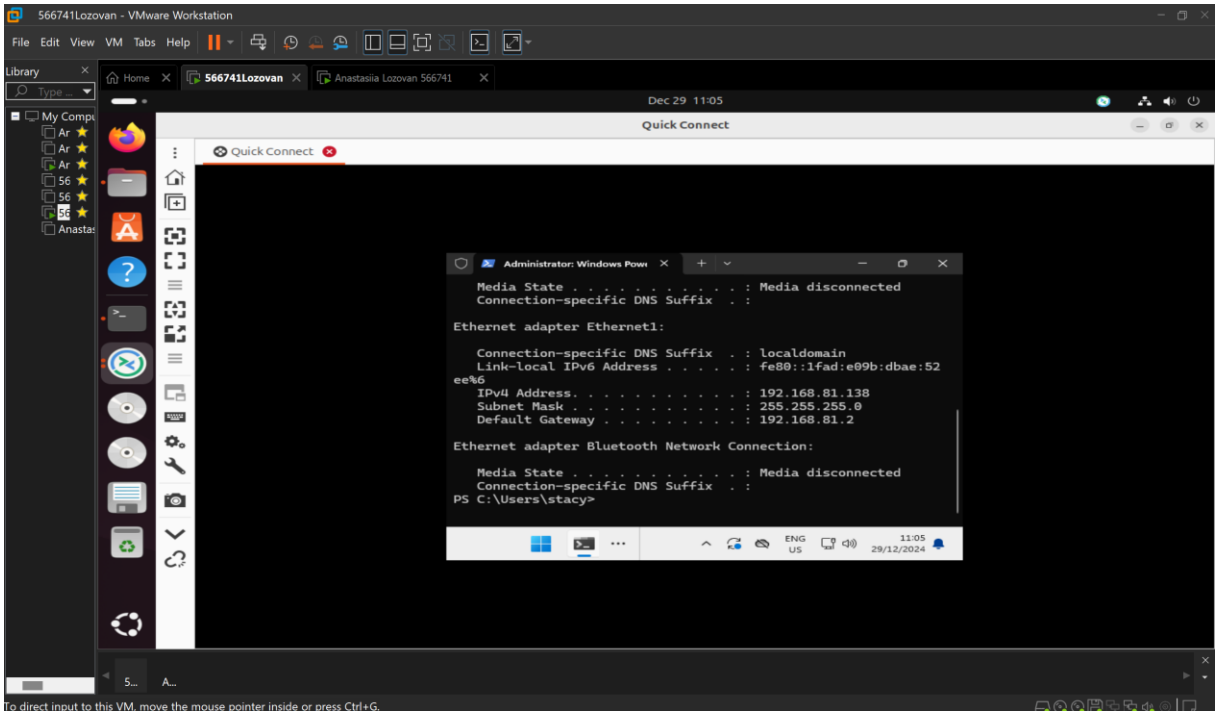


Screenshot successful execution SCP command:



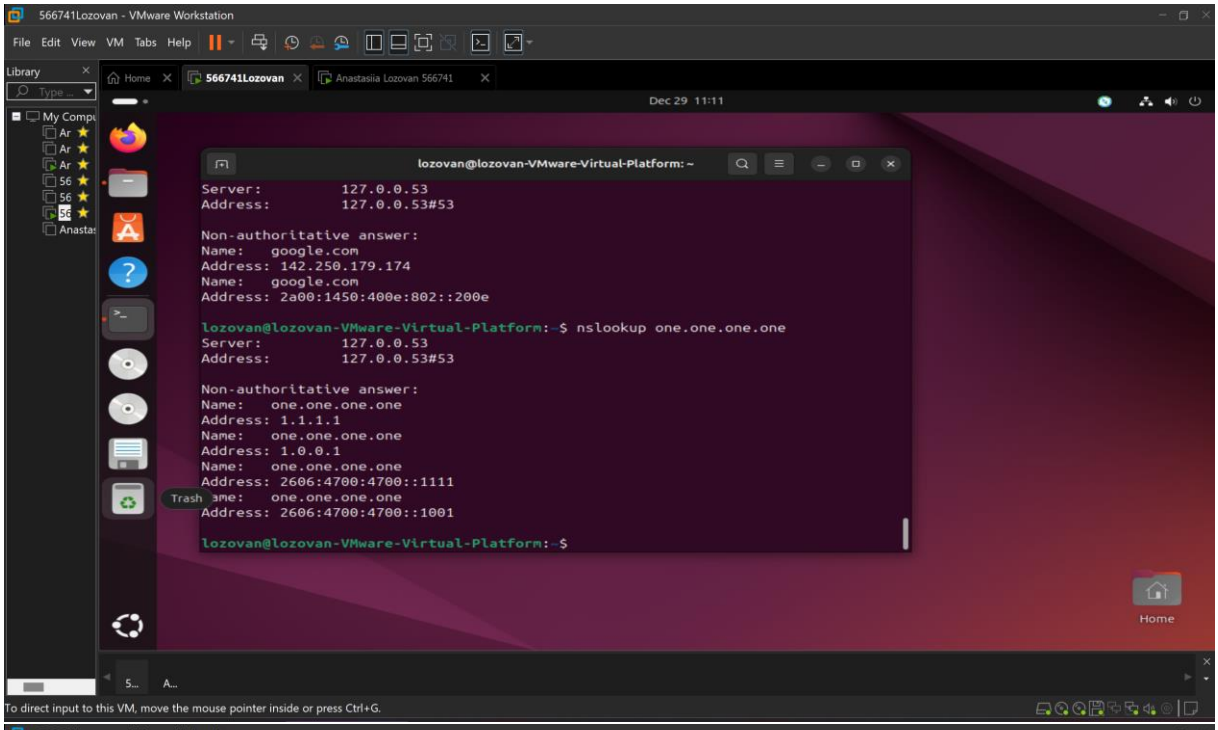


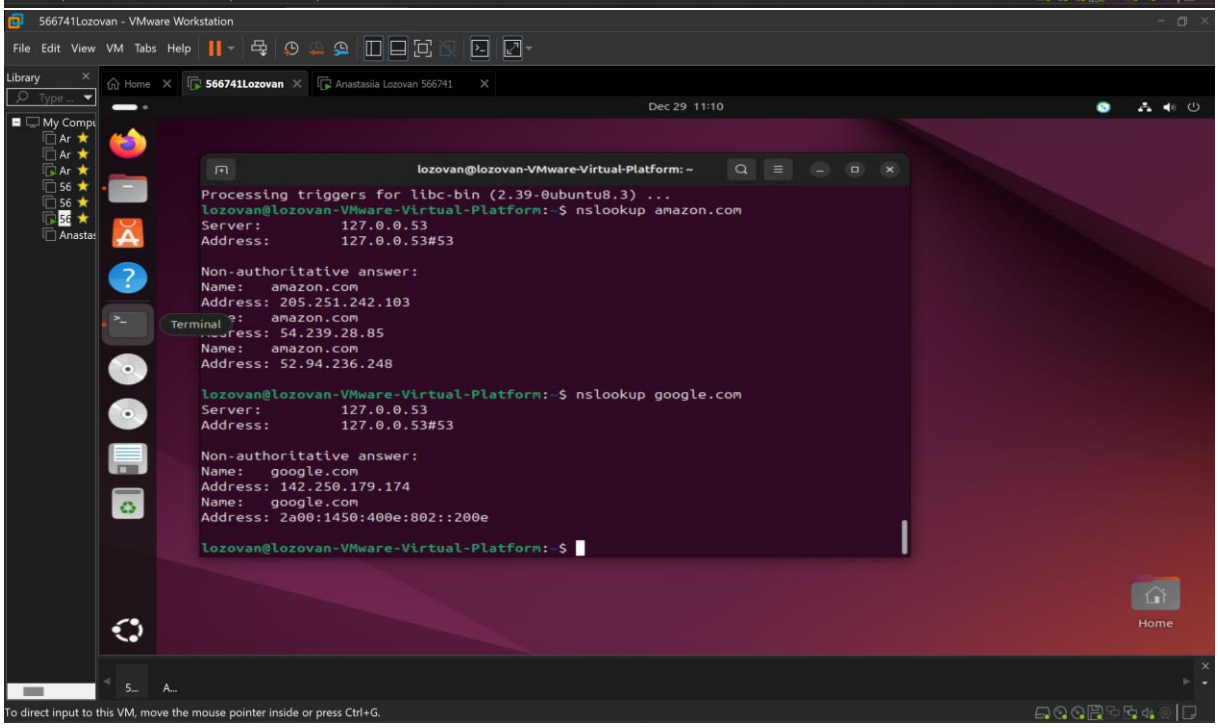
Screenshot remmina:

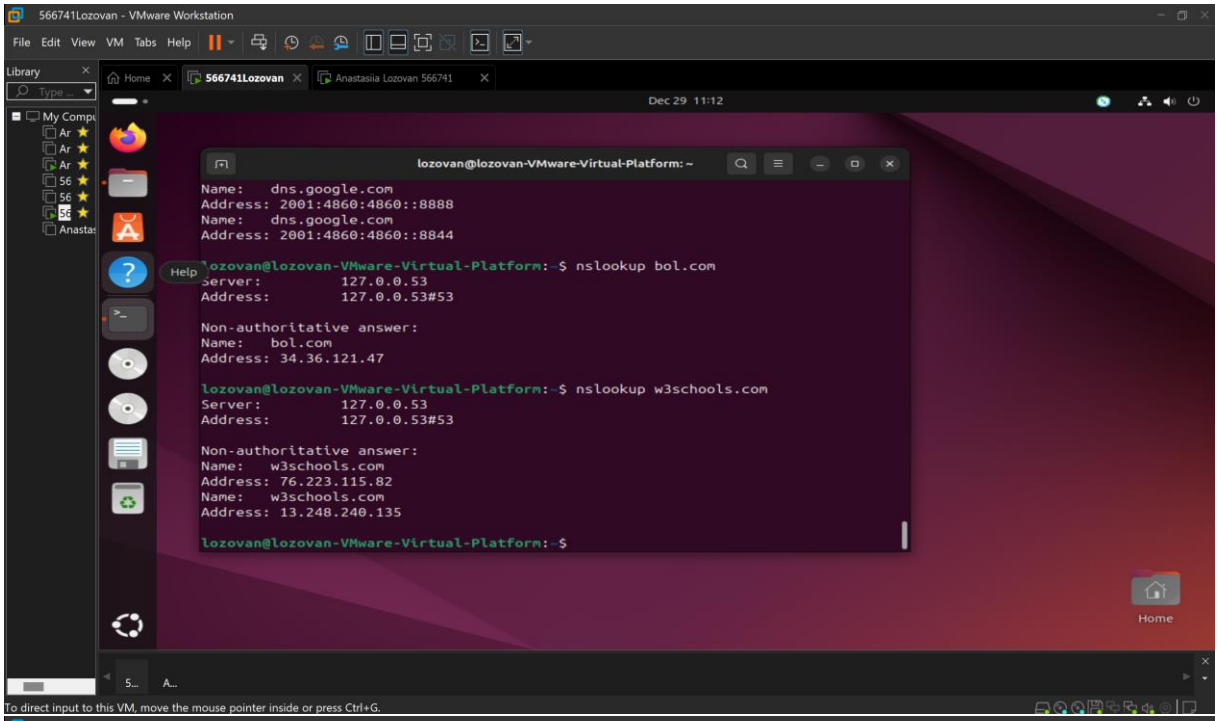


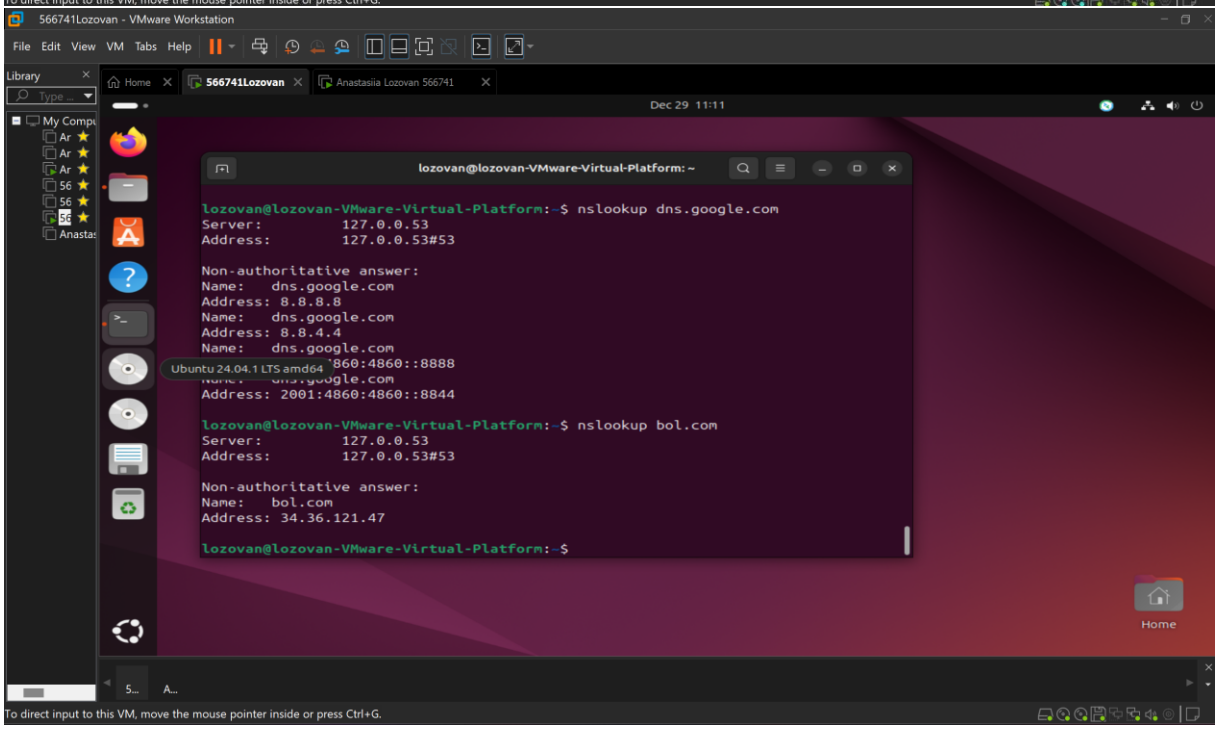
Assignment 6.2: IP addresses websites

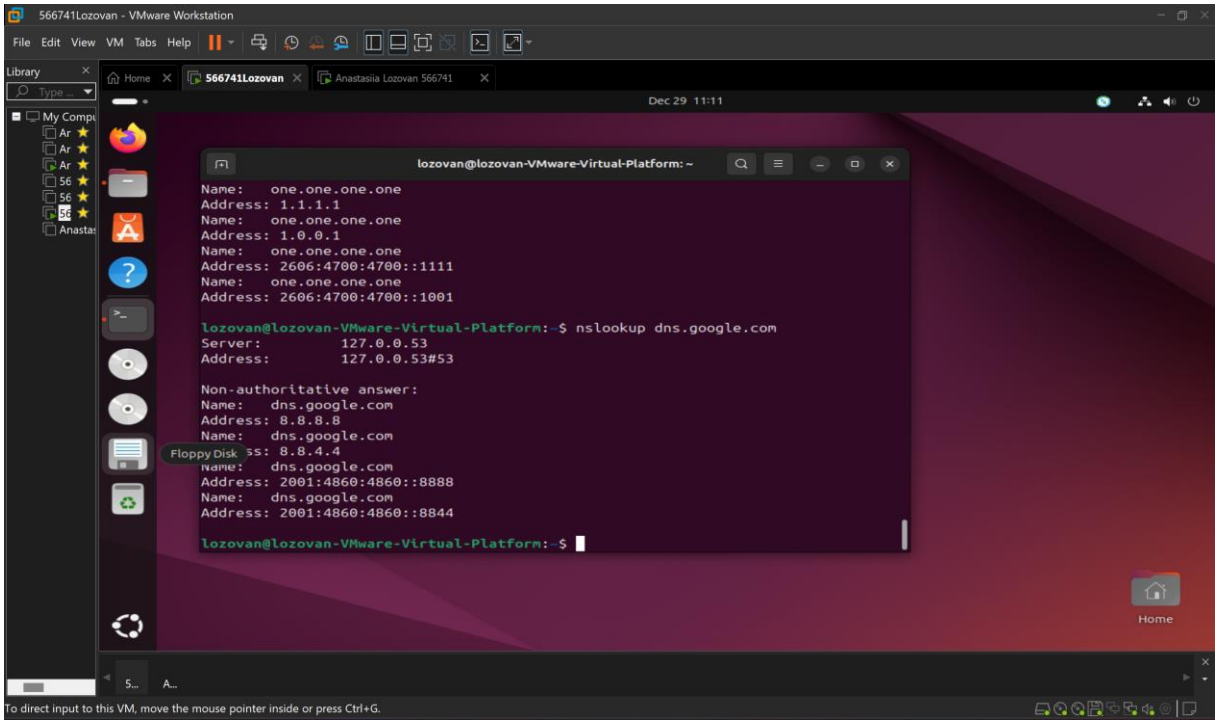
Relevant screenshots nslookup command:


566741Lozovan - VMware Workstation
File Edit View VM Tabs Help
Library
Home
566741Lozovan
Anastasiia Lozovan 566741
Dec 29 11:11
lozovan@lozovan-VMware-Virtual-Platform: ~
Server: 127.0.0.53
Address: 127.0.0.53#53
Non-authoritative answer:
Name: google.com
Address: 142.250.179.174
Name: google.com
Address: 2a00:1450:400e:802::200e
lozovan@lozovan-VMware-Virtual-Platform:~$ nslookup one.one.one.one
Server: 127.0.0.53
Address: 127.0.0.53#53
Non-authoritative answer:
Name: one.one.one.one
Address: 1.1.1.1
Name: one.one.one.one
Address: 1.0.0.1
Name: one.one.one.one
Address: 2606:4700:4700::1111
Name: one.one.one.one
Address: 2606:4700:4700::1001
lozovan@lozovan-VMware-Virtual-Platform:~$
Trash
Home
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

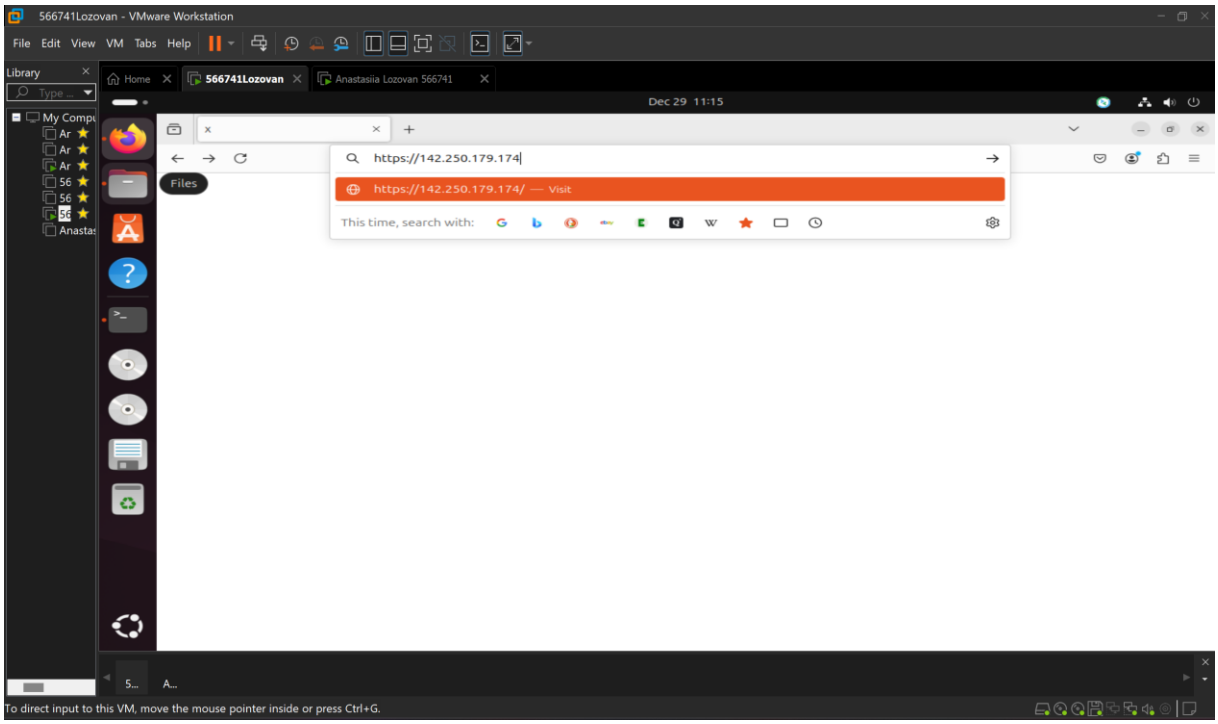

566741Lozovan - VMware Workstation
File Edit View VM Tabs Help
Library
Home
566741Lozovan
Anastasiia Lozovan 566741
Dec 29 11:10
lozovan@lozovan-VMware-Virtual-Platform: ~
Processing triggers for libc-bin (2.39-0ubuntu8.3) ...
lozovan@lozovan-VMware-Virtual-Platform:~$ nslookup amazon.com
Server: 127.0.0.53
Address: 127.0.0.53#53
Non-authoritative answer:
Name: amazon.com
Address: 205.251.242.103
Name: amazon.com
Address: 54.239.28.85
Name: amazon.com
Address: 52.94.236.248
lozovan@lozovan-VMware-Virtual-Platform:~$ nslookup google.com
Server: 127.0.0.53
Address: 127.0.0.53#53
Non-authoritative answer:
Name: google.com
Address: 142.250.179.174
Name: google.com
Address: 2a00:1450:400e:802::200e
lozovan@lozovan-VMware-Virtual-Platform:~$
Terminal
Home
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

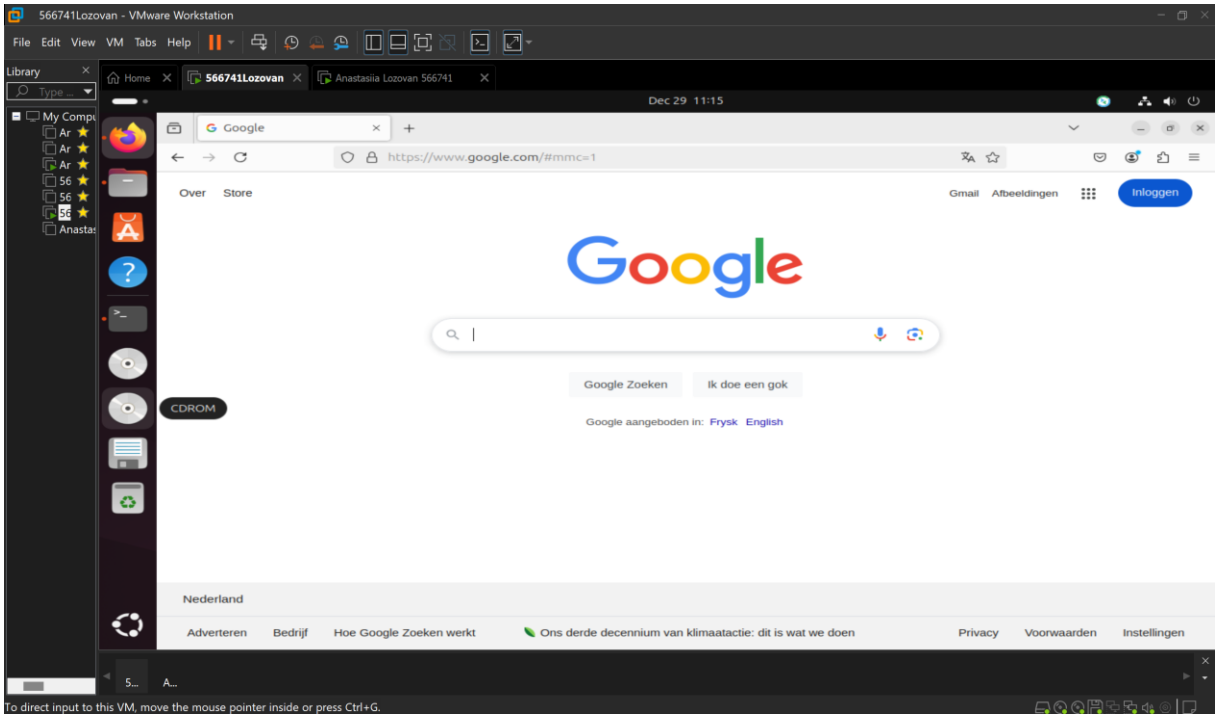

566741Lozovan - VMware Workstation
File Edit View VM Tabs Help
Library
Home
566741Lozovan
Anastasiia Lozovan 566741
Dec 29 11:12
lozovan@lozovan-VMware-Virtual-Platform: ~
Name: dns.google.com
Address: 2001:4860:4860::8888
Name: dns.google.com
Address: 2001:4860:4860::8844
lozovan@lozovan-VMware-Virtual-Platform:~$ nslookup bol.com
Server: 127.0.0.53
Address: 127.0.0.53#53
Non-authoritative answer:
Name: bol.com
Address: 34.36.121.47
lozovan@lozovan-VMware-Virtual-Platform:~$ nslookup w3schools.com
Server: 127.0.0.53
Address: 127.0.0.53#53
Non-authoritative answer:
Name: w3schools.com
Address: 76.223.115.82
Name: w3schools.com
Address: 13.248.240.135
lozovan@lozovan-VMware-Virtual-Platform:~$
Help
Home
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.


566741Lozovan - VMware Workstation
File Edit View VM Tabs Help
Library
Home
566741Lozovan
Anastasiia Lozovan 566741
Dec 29 11:11
lozovan@lozovan-VMware-Virtual-Platform: ~
lozovan@lozovan-VMware-Virtual-Platform:~$ nslookup dns.google.com
Server: 127.0.0.53
Address: 127.0.0.53#53
Non-authoritative answer:
Name: dns.google.com
Address: 8.8.8.8
Name: dns.google.com
Address: 8.8.4.4
Name: dns.google.com
Address: 2001:4860:4860::8844
lozovan@lozovan-VMware-Virtual-Platform:~$ nslookup bol.com
Server: 127.0.0.53
Address: 127.0.0.53#53
Non-authoritative answer:
Name: bol.com
Address: 34.36.121.47
lozovan@lozovan-VMware-Virtual-Platform:~$
Ubuntu 24.04.1 LTS amd64
Home
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.



Screenshot website visit via IP address:





Assignment 6.3: subnetting

How many IP addresses are in this network configuration 192.168.110.128/25?

The network `192.168.110.128/25` has **128 total IP addresses**. The `/25` part tells us that 25 bits are reserved for identifying the network, leaving 7 bits for host devices. With 7 bits, we calculate the total number of addresses like this: 2equation7=128 total addresses.

What is the usable IP range to hand out to the connected computers?

Out of the 128 total addresses, two are special: The network address (192.168.110.128) identifies the network itself. The broadcast address (192.168.110.255) is used to send messages to all devices in the network. This leaves us with 126 usable addresses for devices like computers, printers, or other equipment. The usable range starts from 192.168.110.129 and goes up to 192.168.110.254.

Check your two previous answers with this calculator:
https://www.calculator.net/ip-subnet-calculator.html

Calculator.net FINANCIAL FITNESS & HEALTH

home / other / ip subnet calculator

IP Subnet Calculator

This calculator returns a variety of information regarding Internet Protocol version 4 (IPv4) and IPv6 subnets including possible network addresses, usable host ranges, subnet mask, and IP class, among others.

IPv4 Subnet Calculator

Result

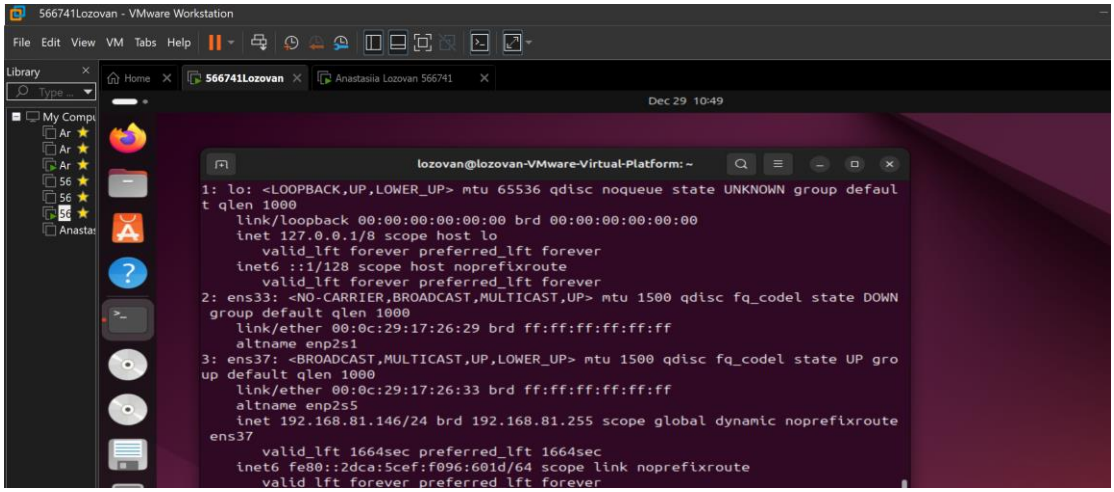
IP Address:	192.168.110.128
Network Address:	192.168.110.128
Usable Host IP Range:	192.168.110.129 - 192.168.110.254
Broadcast Address:	192.168.110.255
Total Number of Hosts:	128
Number of Usable Hosts:	126
Subnet Mask:	255.255.255.128
Wildcard Mask:	0.0.0.127
Binary Subnet Mask:	11111111.11111111.11111111.10000000
IP Class:	C
CIDR Notation:	/25
IP Type:	Private
Short:	192.168.110.128 /25
Binary ID:	11000000101010000110111010000000
Integer ID:	3232263808
Hex ID:	0xc0a86e80
in-addr.arpa:	128.110.168.192.in-addr.arpa
IPv4 Mapped Address:	::ffff:c0a8.6e80
6to4 Prefix:	2002:c0a8.6e80::/48

All 2 of the Possible /25 Networks for 192.168.110.*

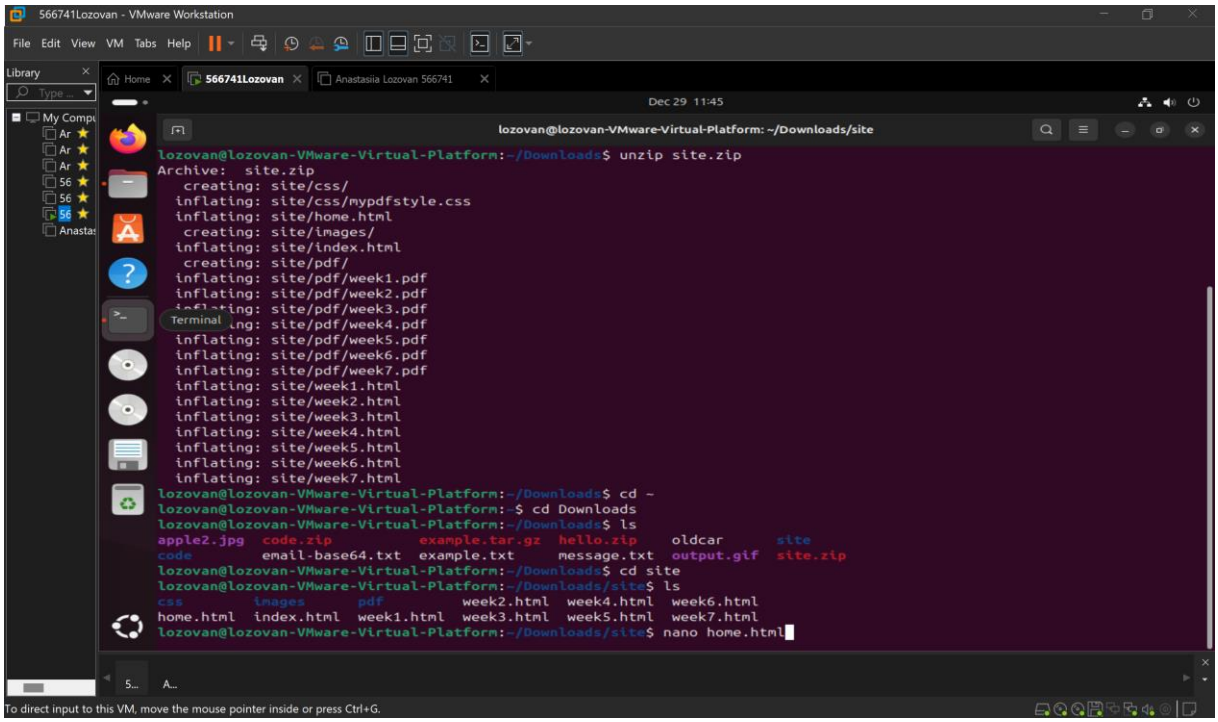
Explain the above calculation in your own words.

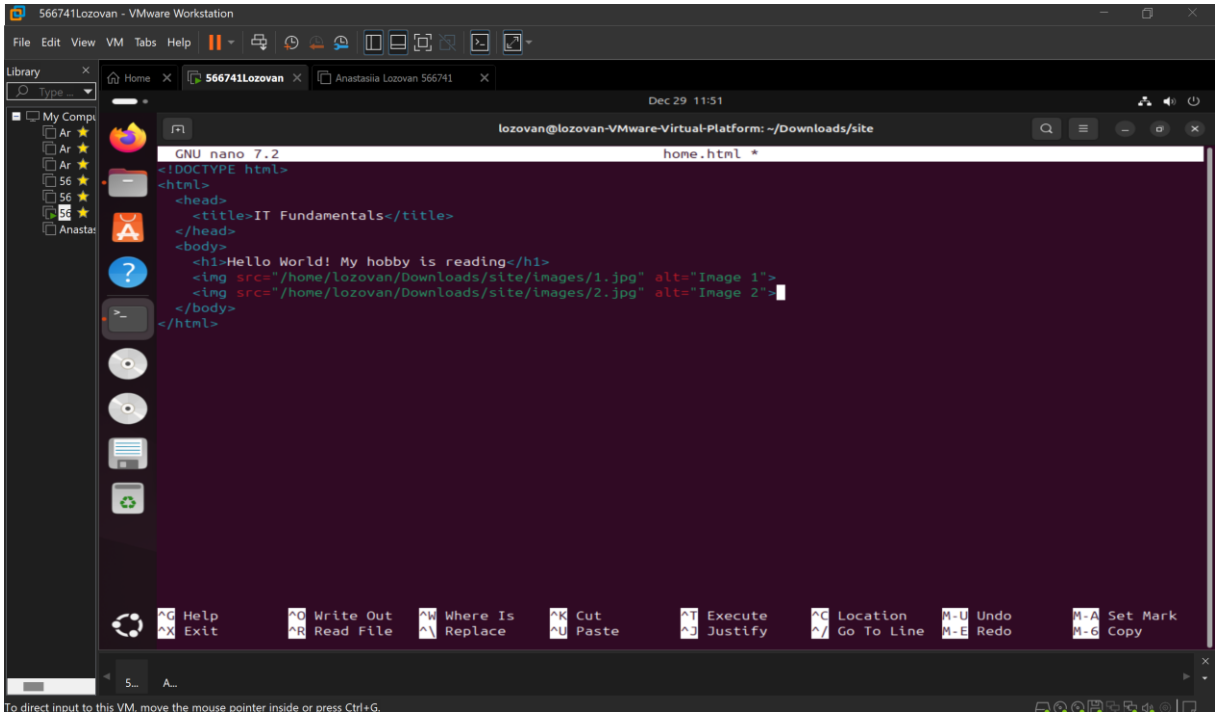
Assignment 6.4: HTML

Screenshot IP address Ubuntu VM:

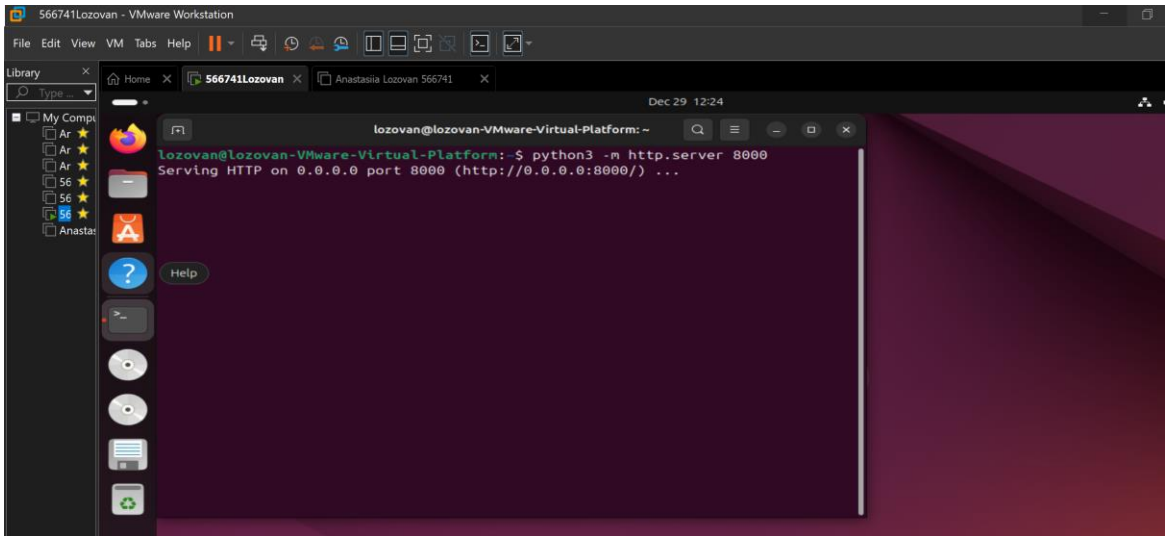


Screenshot of Site directory contents:

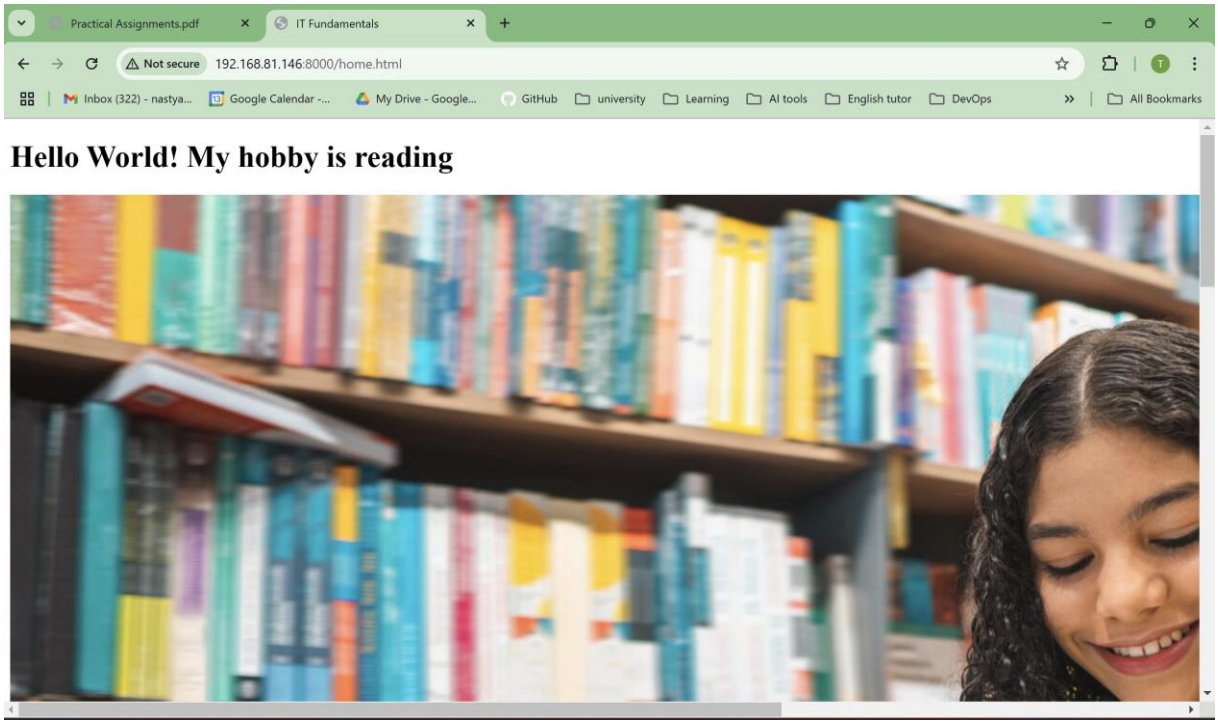




Screenshot python3 webserver command:



Screenshot web browser visits your site



Bonus point assignment – week 6

Remember that bitwise java application you've made in week 2? Expand that application so that you can also calculate a network segment as explained in the PowerPoint slides of week 6. Use the bitwise & AND operator. You need to be able to input two Strings. An IP address and a subnet.

IP: 192.168.1.100 and subnet: 255.255.255.224 for /27

```
Example: 192.168.1.100/27
Calculate the network segment
IP Address:   11000000.10101000.00000001.01100100
Subnet Mask:  11111111.11111111.11111111.11100000
-----------------------------------------------
Network Addr: 11000000.10101000.00000001.01100000

This gives 192.168.1.96 in decimal as the network address.
For a /27 subnet, each segment (or subnet) has 32 IP addresses (2⁵).
The range of this network segment is from 192.168.1.96 to 192.168.1.127.
```

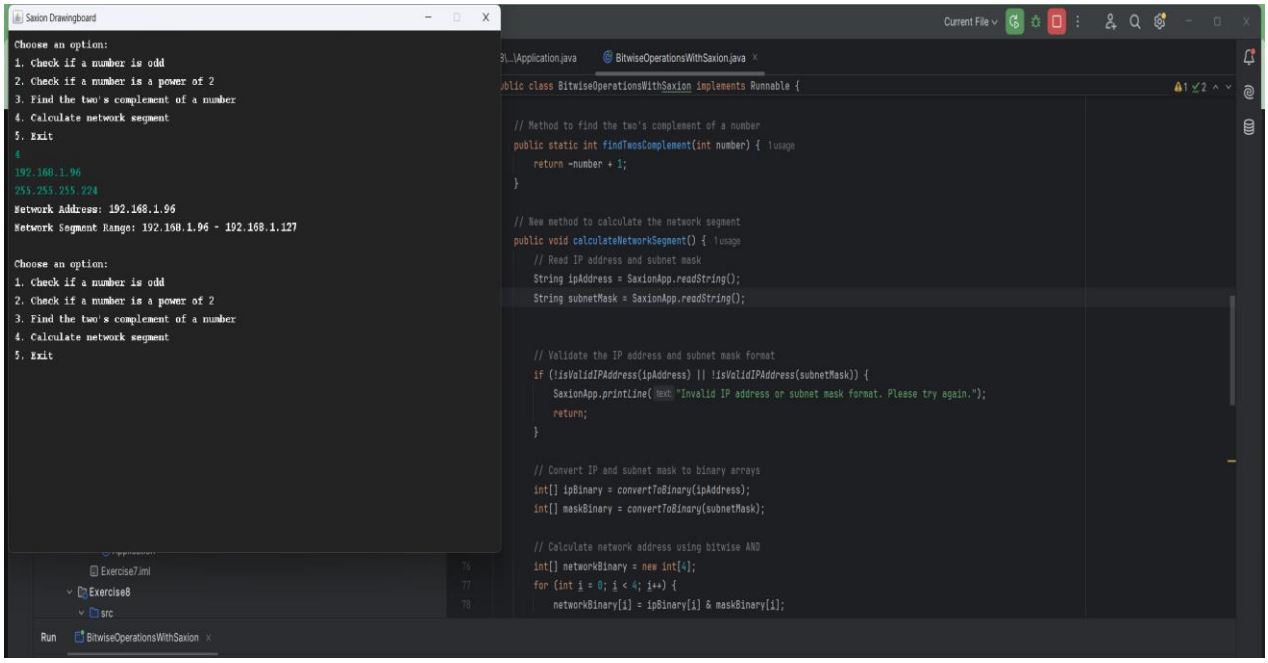
Paste source code here, with a screenshot of a working application.

```
import nl.saxion.app.SaxionApp;

public class BitwiseOperationsWithSaxion implements Runnable {
    public static void main(String[] args) {
        SaxionApp.start(new BitwiseOperationsWithSaxion(), 800, 600); //
Start SaxionApp with a specific window size
    }
```

```java
    @Override
    public void run() {
        while (true) {
            SaxionApp.printLine("Choose an option:");
            SaxionApp.printLine("1. Check if a number is odd");
            SaxionApp.printLine("2. Check if a number is a power of 2");
            SaxionApp.printLine("3. Find the two's complement of a
number");
            SaxionApp.printLine("4. Calculate network segment");
            SaxionApp.printLine("5. Exit");
            int choice = SaxionApp.readInt("Enter your choice:");

            switch (choice) {
                case 1 -> {
                    int number = SaxionApp.readInt("Enter a number:");
                    SaxionApp.printLine("Is the number odd? " +
isOdd(number));
                }
                case 2 -> {
                    int number = SaxionApp.readInt("Enter a number:");
                    SaxionApp.printLine("Is the number a power of 2? " +
isPowerOfTwo(number));
                }
                case 3 -> {
                    int number = SaxionApp.readInt("Enter a number:");
                    SaxionApp.printLine("Two's complement: " +
findTwosComplement(number));
                }
                case 4 -> calculateNetworkSegment();
                case 5 -> {
                    SaxionApp.printLine("Exiting...");
                    System.exit(0); // Exit the application
                }
                default -> SaxionApp.printLine("Invalid choice. Please try
again.");
            }
            SaxionApp.printLine(); // Empty line for readability
        }
    }


    public static boolean isOdd(int number) {
        return (number & 1) == 1;
    }


    public static boolean isPowerOfTwo(int number) {
        return (number > 0) && ((number & (number - 1)) == 0);
    }
```

```java
    public static int findTwosComplement(int number) {
        return ~number + 1;
    }


    public void calculateNetworkSegment() {
        // Read IP address and subnet mask
        String ipAddress = SaxionApp.readString();
        String subnetMask = SaxionApp.readString();


        if (!isValidIPAddress(ipAddress) || !isValidIPAddress(subnetMask))
{
            SaxionApp.printLine("Invalid IP address or subnet mask format.
Please try again.");
            return;
        }


        int[] ipBinary = convertToBinary(ipAddress);
        int[] maskBinary = convertToBinary(subnetMask);


        int[] networkBinary = new int[4];
        for (int i = 0; i < 4; i++) {
            networkBinary[i] = ipBinary[i] & maskBinary[i];
        }


        String networkAddress = convertToDecimal(networkBinary);


        String[] range = calculateRange(networkAddress, subnetMask);


        SaxionApp.printLine("Network Address: " + networkAddress);
        SaxionApp.printLine("Network Segment Range: " + range[0] + " - " +
range[1]);
    }


    public static boolean isValidIPAddress(String ip) {
        String ipPattern = "^((25[0-5]|2[0-4][0-9]|[0-1]?[0-9][0-
9]?)\\.){3}(25[0-5]|2[0-4][0-9]|[0-1]?[0-9][0-9]?)$";
        return ip.matches(ipPattern);
    }
```

```java
    private static int[] convertToBinary(String address) {
        String[] parts = address.split("\\.");
        int[] binary = new int[4];
        for (int i = 0; i < 4; i++) {
            binary[i] = Integer.parseInt(parts[i]);
        }
        return binary;
    }


    private static String convertToDecimal(int[] binary) {
        return binary[0] + "." + binary[1] + "." + binary[2] + "." +
binary[3];
    }


   private static String[] calculateRange(String networkAddress, String
subnetMask) {
        String[] networkParts = networkAddress.split("\\.");
        String[] maskParts = subnetMask.split("\\.");

        int networkStart = Integer.parseInt(networkParts[3]) &
Integer.parseInt(maskParts[3]);
        int broadcastEnd = networkStart | (~Integer.parseInt(maskParts[3])
& 0xFF);

        String start = networkParts[0] + "." + networkParts[1] + "." +
networkParts[2] + "." + networkStart;
        String end = networkParts[0] + "." + networkParts[1] + "." +
networkParts[2] + "." + broadcastEnd;

        return new String[]{start, end};
    }
}
```



Ready? Save this file and export it as a pdf file with the name: **week6.pdf**