

Template Week 6 – Networking

Student number: 575933

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:

Screenshot website visit via IP address:

Assignment 6.3: subnetting

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

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

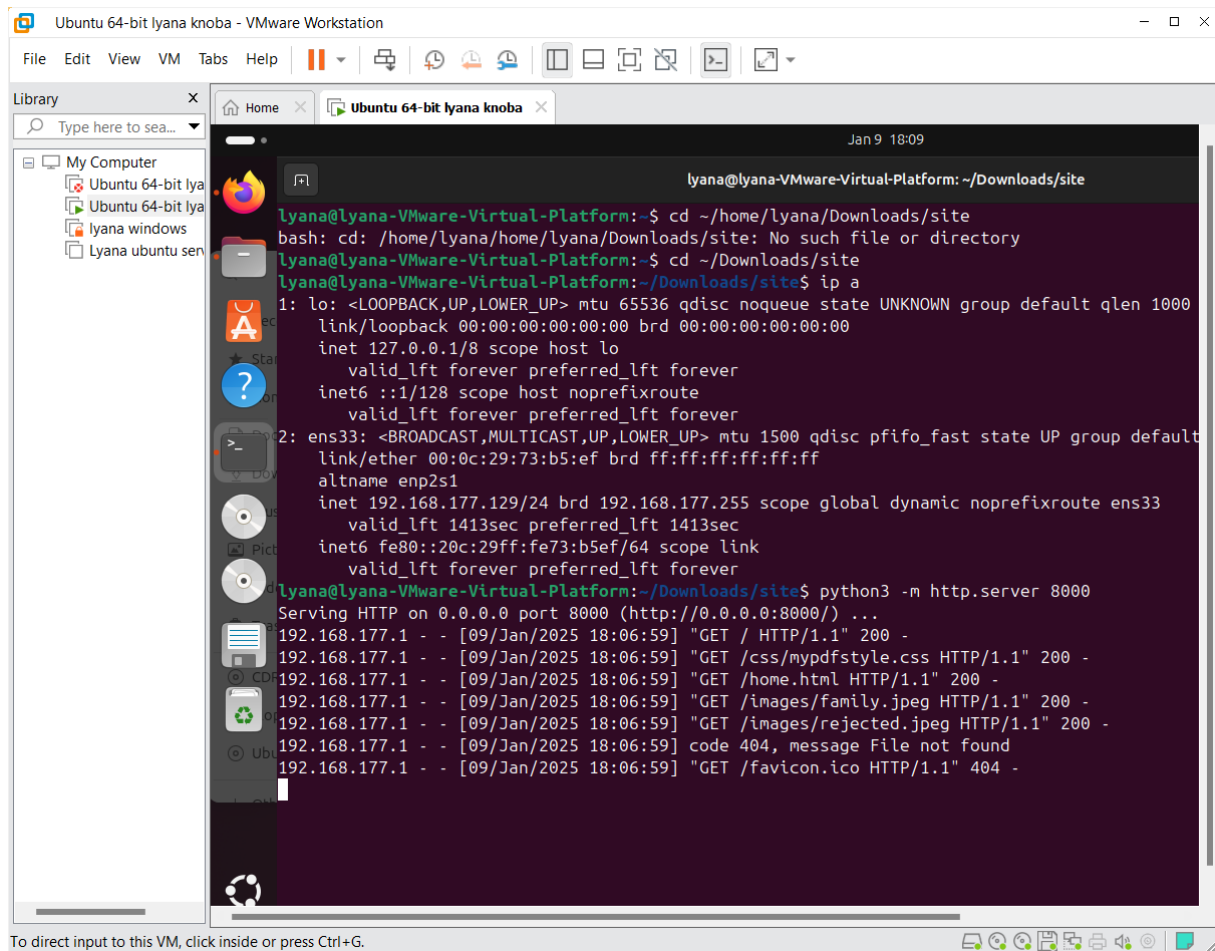
Check your two previous answers with this calculator:

<https://www.calculator.net/ip-subnet-calculator.html>

Explain the above calculation in your own words.

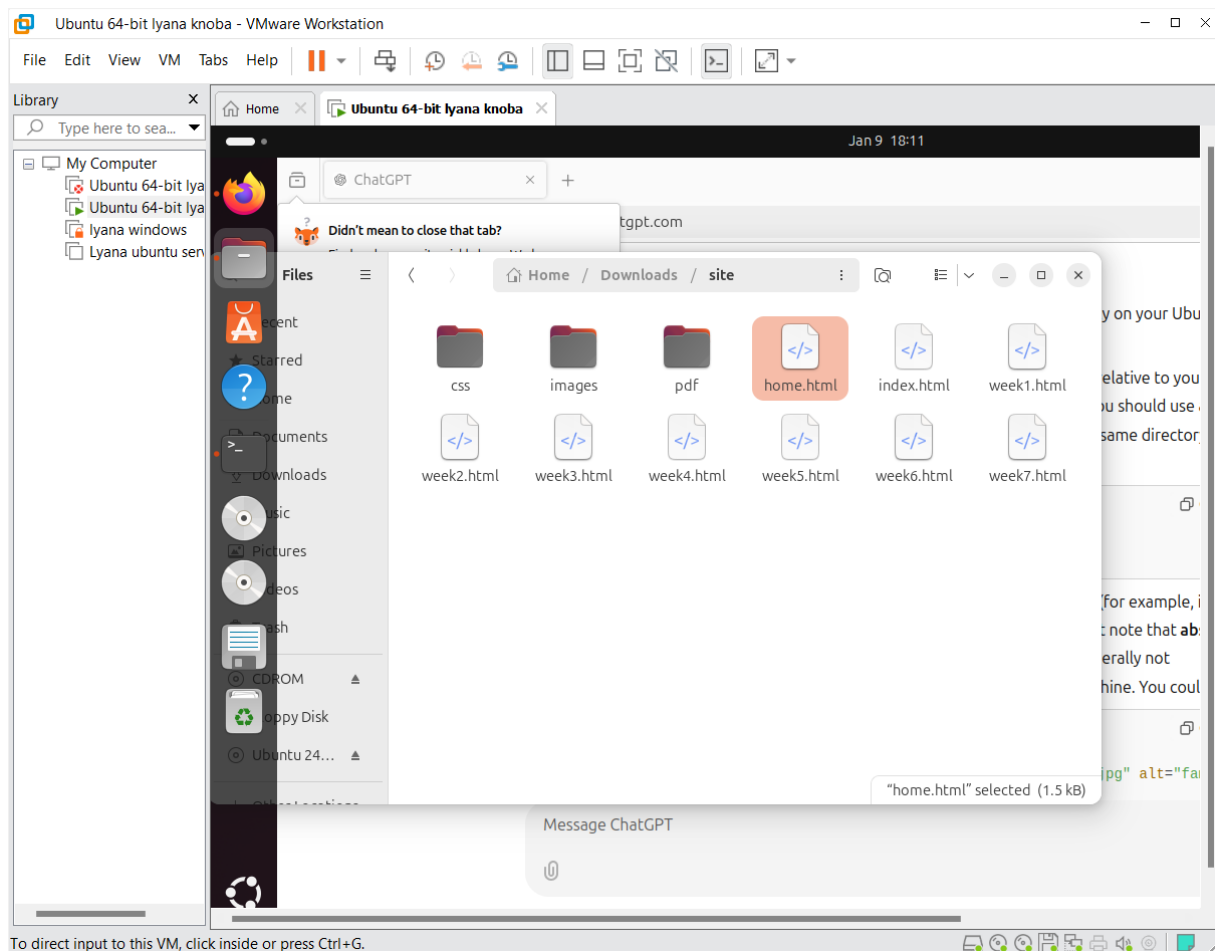
Assignment 6.4: HTML

Screenshot IP address Ubuntu VM:

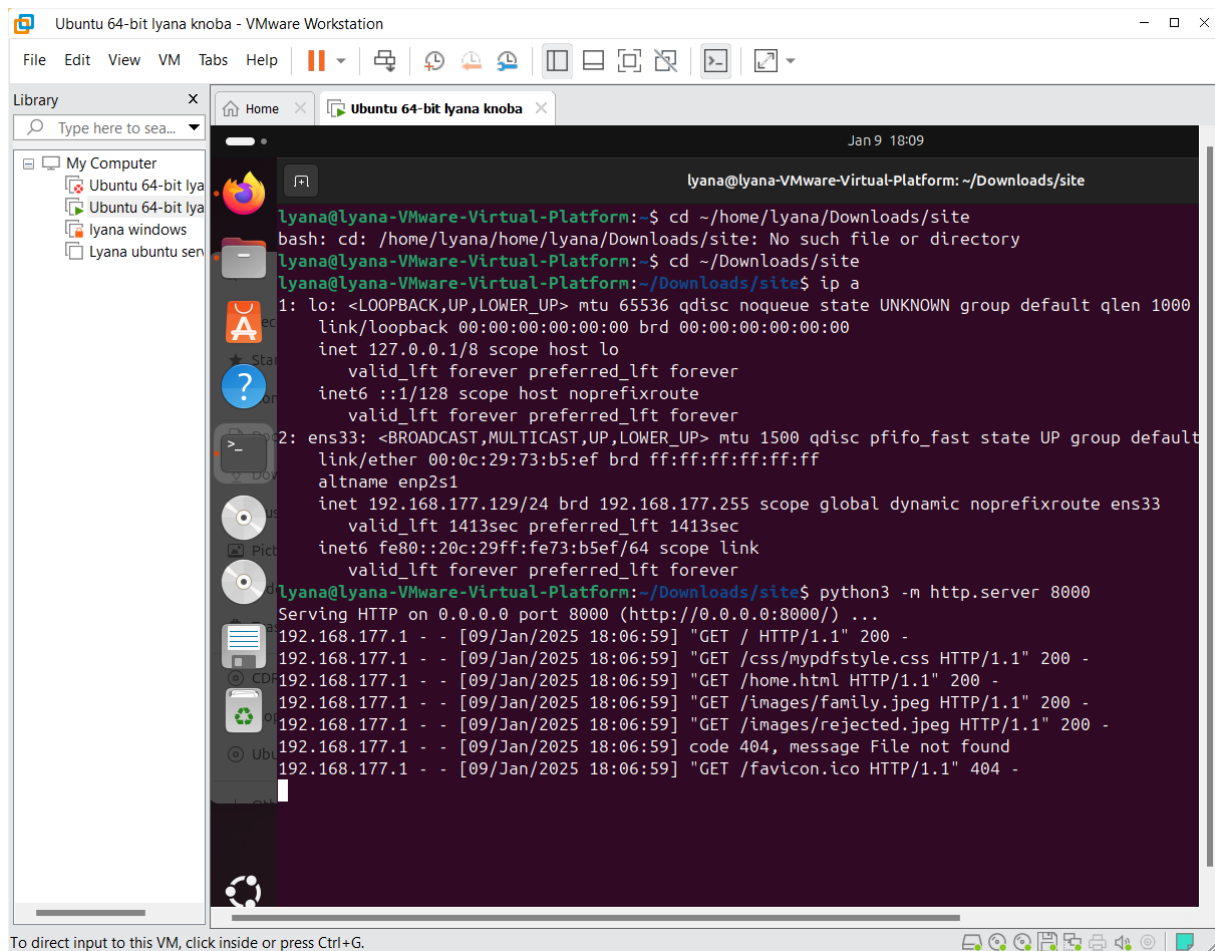


```
lyana@lyana-VMware-Virtual-Platform: ~/Downloads/site
lyana@lyana-VMware-Virtual-Platform:~$ cd ~/home/lyana/Downloads/site
bash: cd: /home/lyana/home/lyana/Downloads/site: No such file or directory
lyana@lyana-VMware-Virtual-Platform:~$ cd ~/Downloads/site
lyana@lyana-VMware-Virtual-Platform:~/Downloads/site$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default
    link/ether 00:0c:29:73:b5:ef brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.177.129/24 brd 192.168.177.255 scope global dynamic noprefixroute ens33
        valid_lft 1413sec preferred_lft 1413sec
    inet6 fe80::20c:29ff:fe73:b5ef/64 scope link
        valid_lft forever preferred_lft forever
lyana@lyana-VMware-Virtual-Platform:~/Downloads/site$ python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
192.168.177.1 - - [09/Jan/2025 18:06:59] "GET / HTTP/1.1" 200 -
192.168.177.1 - - [09/Jan/2025 18:06:59] "GET /css/mypdfstyle.css HTTP/1.1" 200 -
192.168.177.1 - - [09/Jan/2025 18:06:59] "GET /home.html HTTP/1.1" 200 -
192.168.177.1 - - [09/Jan/2025 18:06:59] "GET /images/family.jpeg HTTP/1.1" 200 -
192.168.177.1 - - [09/Jan/2025 18:06:59] "GET /images/rejected.jpeg HTTP/1.1" 200 -
192.168.177.1 - - [09/Jan/2025 18:06:59] "code 404, message File not found"
192.168.177.1 - - [09/Jan/2025 18:06:59] "GET /favicon.ico HTTP/1.1" 404 -
```

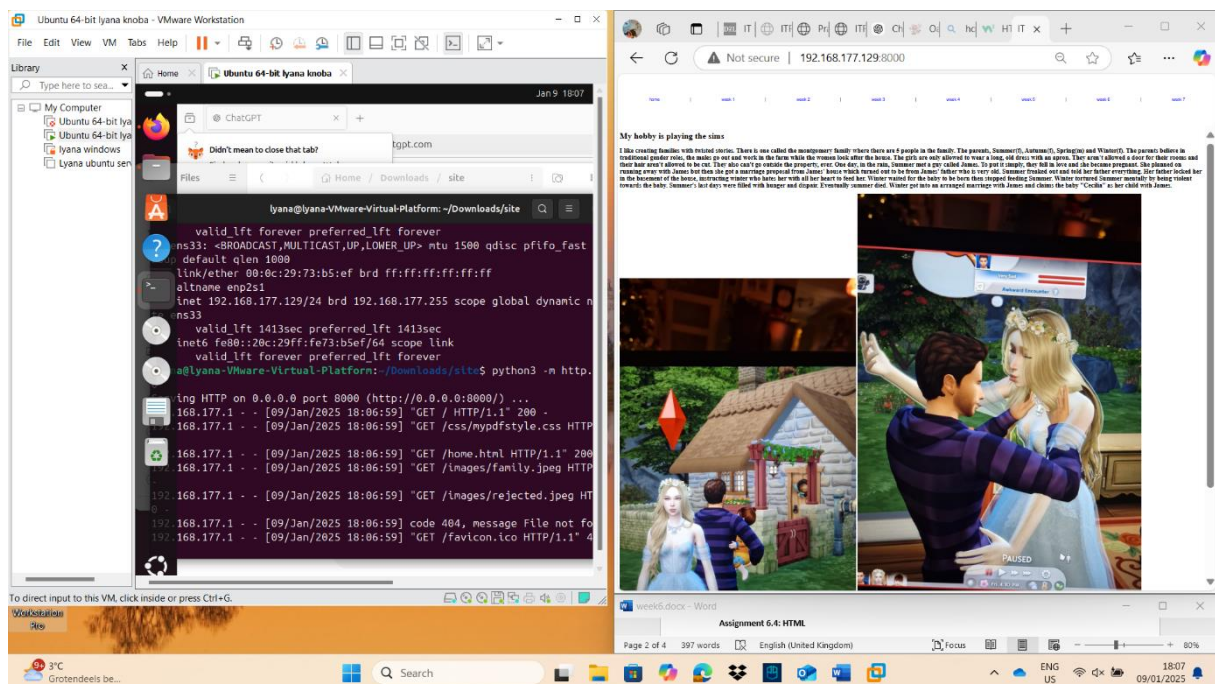
Screenshot of Site directory contents:



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 (2^5).

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 Application implements Runnable {
```

```
    public static void main(String[] args) {  
        SaxionApp.start(new Application(), 640, 200);  
    }
```

```
    public void run() {  
        // Prompt user for IP address and subnet mask  
        SaxionApp.print("Give me an IP");  
        String ip = SaxionApp.readString();  
        SaxionApp.print("Give me a subnet");  
        String subnet = SaxionApp.readString();
```

```
        // Split the IP and subnet mask into individual parts (octets)  
        String[] ipParts = ip.split("\\.");  
        String[] subnetParts = subnet.split("\\.");
```

```
        // Prepare the network address  
        String networkAddress = "";
```

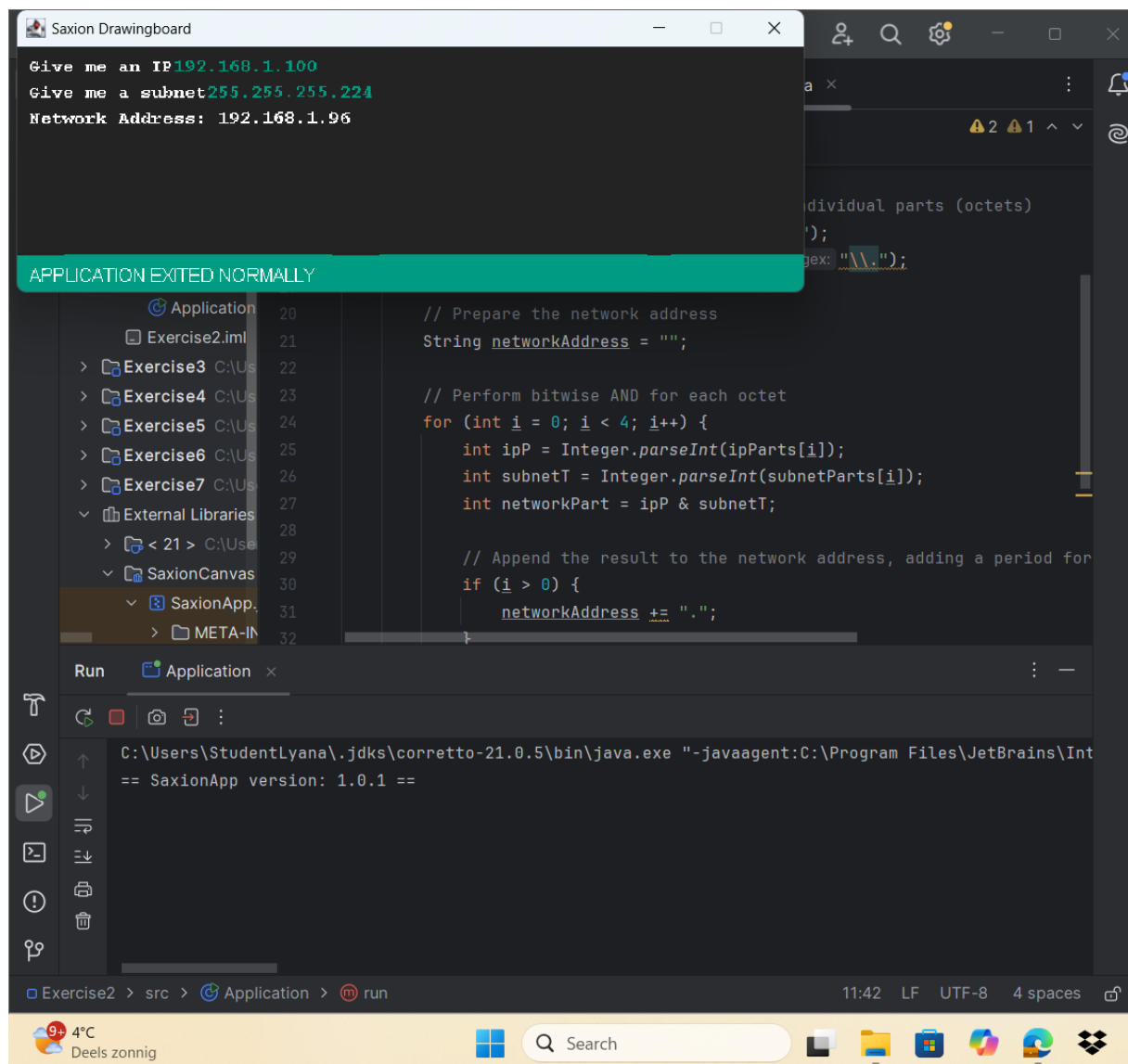
```

// Perform bitwise AND for each octet
for (int i = 0; i < 4; i++) {
    int ipP = Integer.parseInt(ipParts[i]);
    int subnetT = Integer.parseInt(subnetParts[i]);
    int networkPart = ipP & subnetT;

    // Append the result to the network address, adding a period for each part after the first
    if (i > 0) {
        networkAddress += ".";
    }
    networkAddress += networkPart;
}

// Output the calculated network address
SaxionApp.print("Network Address: " + networkAddress);
}
}

```



Ready? Save this file and export it as a pdf file with the name: [week6.pdf](#)