This document provides detailed instructions for Part 1 of Final Project I for the course 420-635-AB-Network Installation and Administration I. The primary objectives of this assignment are setting up a basic web server using Apache on a Linux operating system and creating a simple website with HTML. The process involves installing and configuring the Apache web server, managing its service, understanding the fundamental structure of HTML documents, and utilizing key HTML tags to create content and hyperlinks.

Project Part I

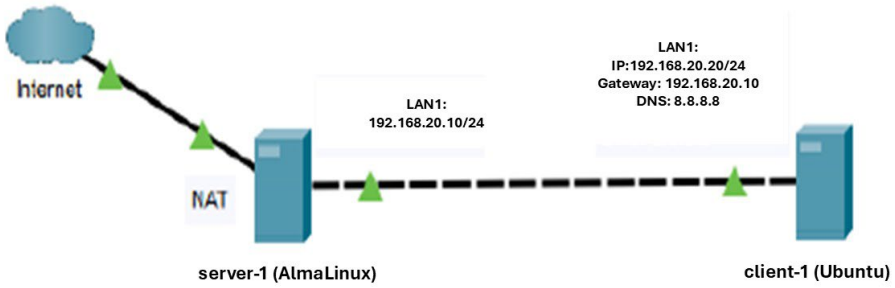
420-635-AB-Network Installation and Administration I

Teacher: Antoine Tohme
Student: Monica Perez Mata
Student id : 2498056

Table of Contents

1 Topology

The network topology to be used is the same as used in previous assignments



Server-1 with Alma-Linux is to be used as Apache server

2 Installing the Apache Web Server on Alma Linux

2.1 Software updates

The first step is to update the local package index to ensure you have the latest information about available packages.

Connect as root

su –

dnf update -y

```
Error: This command has to be run with superuser privileges (under the root user on most systems).
[mperez@server1 ~]$ su -
Password:
[root@server1 ~]# dnf update -y
Last metadata expiration check: 0:12:15 ago on Tue 15 Apr 2025 11:19:50 PM.
Dependencies resolved.
```

Package	Architecture	Version	Repository	Size
Installing:				
kernel	x86_64	5.14.0-503.35.1.el9_5	baseos	2.0 M
Upgrading:				
bpftool	x86_64	7.4.0-503.35.1.el9_5	baseos	2.8 M
expat	x86_64	2.5.0-3.el9_5.3	baseos	115 k
firefox	x86_64	128.9.0-2.el9_5	appstream	125 M
freetype	x86_64	2.10.4-10.el9_5	baseos	385 k
kernel-headers	x86_64	5.14.0-503.35.1.el9_5	appstream	3.5 M
kernel-tools	x86_64	5.14.0-503.35.1.el9_5	baseos	2.3 M
kernel-tools-libs	x86_64	5.14.0-503.35.1.el9_5	baseos	2.1 M
mesa-dri-drivers	x86_64	24.1.2-3.el9.alma.2	appstream	8.8 M
mesa-filesystem	x86_64	24.1.2-3.el9.alma.2	appstream	10 k
mesa-libEGL	x86_64	24.1.2-3.el9.alma.2	appstream	137 k
mesa-libGL	x86_64	24.1.2-3.el9.alma.2	appstream	169 k
mesa-libgbm	x86_64	24.1.2-3.el9.alma.2	appstream	35 k
mesa-libglapi	x86_64	24.1.2-3.el9.alma.2	appstream	44 k
mesa-libxatracker	x86_64	24.1.2-3.el9.alma.2	appstream	2.1 M
mesa-vulkan-drivers	x86_64	24.1.2-3.el9.alma.2	appstream	11 M
podman	x86_64	4:5.2.2-15.el9_5	appstream	16 M
python3-perf	x86_64	5.14.0-503.35.1.el9_5	appstream	2.1 M
tzdata	noarch	2025b-1.el9	baseos	430 k
tzdata-java	noarch	2025b-1.el9	appstream	145 k
webkit2gtk3	x86_64	2.48.1-1.el9_5	appstream	27 M
webkit2gtk3-jsc	x86_64	2.48.1-1.el9_5	appstream	4.7 M
Installing dependencies:				
kernel-core	x86_64	5.14.0-503.35.1.el9_5	baseos	18 M
kernel-devel	x86_64	5.14.0-503.35.1.el9_5	appstream	18 M
kernel-modules	x86_64	5.14.0-503.35.1.el9_5	baseos	36 M
kernel-modules-core	x86_64	5.14.0-503.35.1.el9_5	baseos	30 M

2.2 Installing the Apache Web Server

To begin the installation, execute the following command in the terminal:

dnf install httpd

```
[root@server1 ~]#
[root@server1 ~]# sudo dnf install httpd -y
Last metadata expiration check: 1:30:51 ago on Tue 15 Apr 2025 11:19:50 PM.
Dependencies resolved.
```

Package	Architecture	Version	Repository	Size
Installing:				
httpd	x86_64	2.4.62-1.el9_5.2	appstream	45 k
Installing dependencies:				
almalinux-logos-httpd	noarch	90.5.1-1.1.el9	appstream	18 k
apr	x86_64	1.7.0-12.el9_3	appstream	122 k
apr-util	x86_64	1.6.1-23.el9	appstream	94 k
apr-util-bdb	x86_64	1.6.1-23.el9	appstream	12 k
httpd-core	x86_64	2.4.62-1.el9_5.2	appstream	1.4 M
httpd-filesystem	noarch	2.4.62-1.el9_5.2	appstream	12 k
httpd-tools	x86_64	2.4.62-1.el9_5.2	appstream	79 k
Installing weak dependencies:				
apr-util-openssl	x86_64	1.6.1-23.el9	appstream	14 k
mod_http2	x86_64	2.0.26-2.el9_4.1	appstream	162 k
mod_lua	x86_64	2.4.62-1.el9_5.2	appstream	58 k

Transaction Summary

Install 11 Packages

2.3 Manage the httpd service

a) Verify the status, by default status is disabled

systemctl status httpd

```
[mperez@server1 ~]$ systemctl status httpd
○ httpd.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
     Active: inactive (dead)
       Docs: man:httpd.service(8)
[mperez@server1 ~]$
```

b) Start httpd to initiate the Apache service

systemctl start httpd

```
[root@server1 ~]# systemctl start httpd
[root@server1 ~]#
```

c) Verify the status, by default status is disabled

systemctl status httpd

Note httpd is **active (running)** but service is disabled

```
[root@server1 ~]# systemctl status httpd
● httpd.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
     Active: active (running) since Wed 2025-04-16 10:43:44 EDT; 23s ago
       Docs: man:httpd.service(8)
   Main PID: 78466 (httpd)
     Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes served/sec:   0 B/sec"
      Tasks: 177 (limit: 22829)
     Memory: 36.2M
        CPU: 638ms
     CGroup: /system.slice/httpd.service
             ├─78466 /usr/sbin/httpd -DFOREGROUND
             ├─78467 /usr/sbin/httpd -DFOREGROUND
             ├─78468 /usr/sbin/httpd -DFOREGROUND
             ├─78499 /usr/sbin/httpd -DFOREGROUND
             └─78523 /usr/sbin/httpd -DFOREGROUND

Apr 16 10:43:28 server1 systemd[1]: Starting The Apache HTTP Server...
Apr 16 10:43:39 server1 httpd[78466]: AH00558: httpd: Could not reliably determine the server's fully qualif
Apr 16 10:43:44 server1 systemd[1]: Started The Apache HTTP Server.
Apr 16 10:43:44 server1 httpd[78466]: Server configured, listening on: port 80
```

d) Check processes used by httpd

See processes running corresponds to the ones seen in systemctl status command above

ps -aux | grep httpd

```
[root@server1 ~]# ps -aux | grep httpd
root       78466  0.0  0.3  21236 11536 ?        Ss   10:43   0:00 /usr/sbin/httpd -DFOREGROUND
apache     78467  0.0  0.1  22968  7152 ?        S    10:43   0:00 /usr/sbin/httpd -DFOREGROUND
apache     78468  0.0  0.5 1572280 19632 ?       Sl   10:43   0:01 /usr/sbin/httpd -DFOREGROUND
apache     78499  0.0  0.4 1441144 15372 ?       Sl   10:43   0:01 /usr/sbin/httpd -DFOREGROUND
apache     78523  0.0  0.4 1441144 15380 ?       Sl   10:43   0:01 /usr/sbin/httpd -DFOREGROUND
root       78958  0.0  0.0 221792  2432 pts/3    S+   12:10   0:00 grep --color=auto httpd
[root@server1 ~]#
```

e) Check which port is open for Apache

netstat -tunap

See from printout httpd is listening on port 80

```
[root@server1 ~]#
[root@server1 ~]# netstat -tunap
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      992/sshd: /usr/sbin
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN      986/cupsd
tcp        0      0 127.0.0.1:6013          0.0.0.0:*               LISTEN      78713/sshd: mperez@
tcp        0      0 127.0.0.1:6012          0.0.0.0:*               LISTEN      78357/sshd: mperez@
tcp        0      0 192.168.186.130:22      192.168.186.1:52776     ESTABLISHED 78711/sshd: mperez
tcp        0      0 192.168.186.130:22      192.168.186.1:52770     ESTABLISHED 78707/sshd: mperez
tcp        0    336 192.168.186.130:22      192.168.186.1:52072     ESTABLISHED 78351/sshd: mperez
tcp        0      0 192.168.186.130:22      192.168.186.1:52077     ESTABLISHED 78355/sshd: mperez
tcp6       0      0 :::22                   :::*                    LISTEN      992/sshd: /usr/sbin
tcp6       0      0 :::80                   :::*                    LISTEN      78466/httpd
tcp6       0      0 ::1:6012                :::*                    LISTEN      78357/sshd: mperez@
tcp6       0      0 ::1:6013                :::*                    LISTEN      78713/sshd: mperez@
tcp6       0      0 ::1:631                 :::*                    LISTEN      986/cupsd
udp        0      0 0.0.0.0:53305           0.0.0.0:*                           763/avahi-daemon: r
udp        0      0 192.168.186.130:68      192.168.186.254:67      ESTABLISHED 964/NetworkManager
udp        0      0 127.0.0.1:323           0.0.0.0:*                           777/chronyd
udp        0      0 0.0.0.0:5353            0.0.0.0:*                           763/avahi-daemon: r
udp6       0      0 :::55244                :::*                                763/avahi-daemon: r
udp6       0      0 ::1:323                 :::*                                777/chronyd
udp6       0      0 :::5353                 :::*                                763/avahi-daemon: r
```

ss -tunapl

See from printout httpd is listening on port 80 and process shown corresponds to some (not all) the processes in printouts above (systemctl status and ps -aux).

```
Netid     State      Recv-Q     Send-Q                  Local Address:Port              Peer Address:Port
tcp       LISTEN     0          511                      *:80                           *:*
Process
users:(("httpd",pid=78523,fd=4),("httpd",pid=78499,fd=4),("httpd",pid=78468,fd=4),("httpd",pid=78466,fd=4))
```

```
[root@server1 ~]# ss -tunap
Netid     State      Recv-Q     Send-Q                  Local Address:Port              Peer Address:Port
Process
udp       UNCONN     0          0                            0.0.0.0:53305                  0.0.0.0:*
 users:(("avahi-daemon",pid=763,fd=14))
udp       ESTAB      0          0                  192.168.186.130%ens160:68              192.168.186.254:67
 users:(("NetworkManager",pid=964,fd=30))
udp       UNCONN     0          0                          127.0.0.1:323                  0.0.0.0:*
 users:(("chronyd",pid=777,fd=5))
udp       UNCONN     0          0                            0.0.0.0:5353                 0.0.0.0:*
 users:(("avahi-daemon",pid=763,fd=12))
udp       UNCONN     0          0                               [::]:55244                   [::]:*
 users:(("avahi-daemon",pid=763,fd=15))
udp       UNCONN     0          0                              [::1]:323                     [::]:*
 users:(("chronyd",pid=777,fd=6))
udp       UNCONN     0          0                               [::]:5353                    [::]:*
 users:(("avahi-daemon",pid=763,fd=13))
tcp       LISTEN     0          128                          0.0.0.0:22                     0.0.0.0:*
 users:(("sshd",pid=992,fd=3))
tcp       LISTEN     0          4096                       127.0.0.1:631                    0.0.0.0:*
 users:(("cupsd",pid=986,fd=8))
tcp       LISTEN     0          128                        127.0.0.1:6013                   0.0.0.0:*
 users:(("sshd",pid=78713,fd=9))
tcp       LISTEN     0          128                        127.0.0.1:6012                   0.0.0.0:*
 users:(("sshd",pid=78357,fd=9))
tcp       ESTAB      0          0                  192.168.186.130:22             192.168.186.1:52776
 users:(("sshd",pid=78716,fd=4),("sshd",pid=78711,fd=4))
tcp       ESTAB      0          0                  192.168.186.130:22             192.168.186.1:52770
 users:(("sshd",pid=78713,fd=4),("sshd",pid=78707,fd=4))
tcp       ESTAB      0          48                 192.168.186.130:22             192.168.186.1:52072
 users:(("sshd",pid=78357,fd=4),("sshd",pid=78351,fd=4))
tcp       ESTAB      0          0                  192.168.186.130:22             192.168.186.1:52077
 users:(("sshd",pid=78360,fd=4),("sshd",pid=78355,fd=4))
tcp       LISTEN     0          128                             [::]:22                      [::]:*
 users:(("sshd",pid=992,fd=4))
tcp       LISTEN     0          511                              *:80                         *:*
 users:(("httpd",pid=78523,fd=4),("httpd",pid=78499,fd=4),("httpd",pid=78468,fd=4),("httpd",pid=78466,fd=4))
tcp       LISTEN     0          128                             [::1]:6012                   [::]:*
 users:(("sshd",pid=78357,fd=8))
tcp       LISTEN     0          128                             [::1]:6013                   [::]:*
 users:(("sshd",pid=78713,fd=8))
```

Not all child processes are responsible for listening on the specified port and ss focuses on active sockets and listening ports.

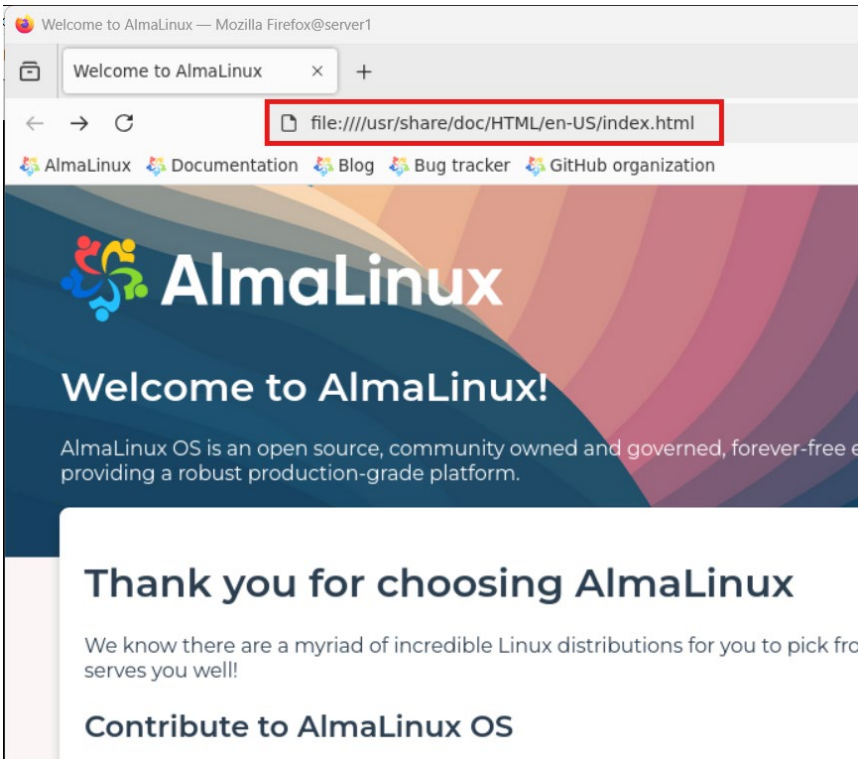
2.4 Verify Apache is working on localhost

2.4.1 Open firefox

firefox &

```
[root@server1 httpd]# firefox &
[1] 79505
```

Per default Alma Linux page opens



2.4.2 Get Alma Linux test page

write **localhost**

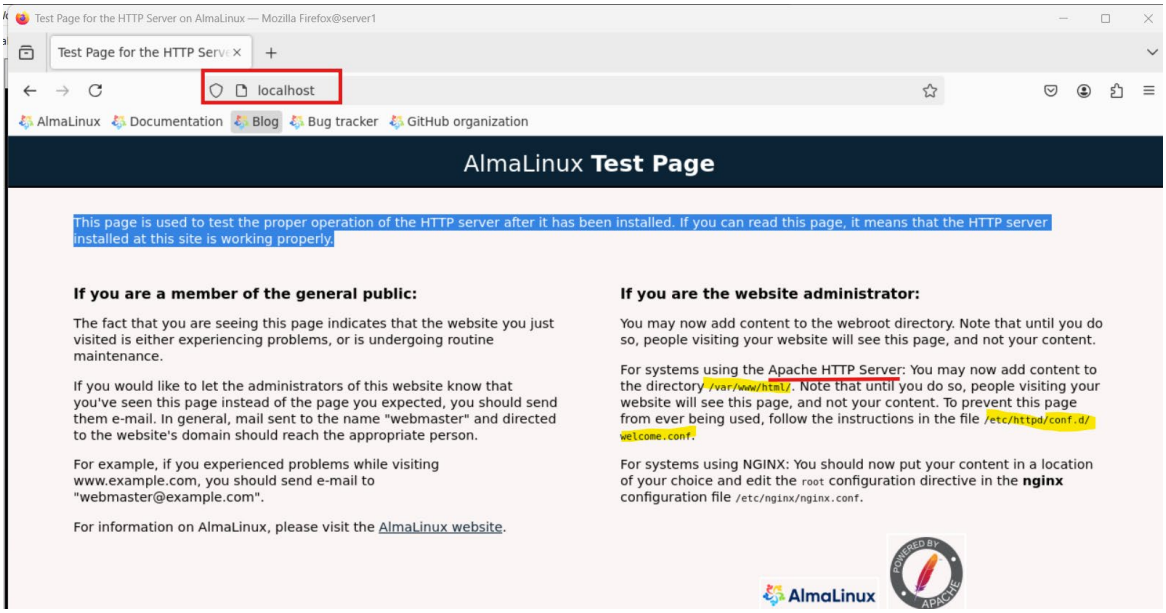
See message ：

This page is used to test the proper operation of the HTTP server after it has been installed. If you can read this page, it means that the HTTP server installed at this site is working properly.

If you are the website administrator:

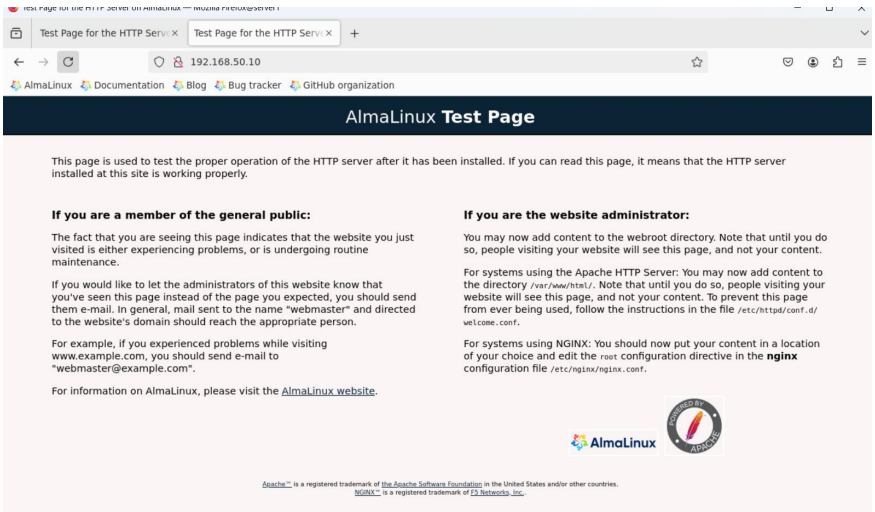
You may now add content to the webroot directory. Note that until you do so, people visiting your website will see this page, and not your content.

*For systems using the Apache HTTP Server: You may now **add content to the directory /var/www/html/**. Note that until you do so, people visiting your website will see this page, and not your content. To prevent this page from ever being used, follow the instructions in the file /etc/httpd/conf.d/welcome.conf.*



You can also access with ip

192.168.50.10



2.5 Firewall

1. Verify existing active zones
   firewall-cmd --get-active-zones

```
[root@server1 httpd]# firewall-cmd --get-active-zones
external
  interfaces: ens160
nm-shared
  interfaces: ens192
[root@server1 httpd]#
```

2. List the firewall sets for the zone

   sudo firewall-cmd --list-all --zone=nm-shared

```
  rich rules:
[root@server1 httpd]# sudo firewall-cmd --list-all --zone=nm-shared
nm-shared (active)
  target: ACCEPT
  icmp-block-inversion: no
  interfaces: ens192
  sources:
  services: dhcp dns ssh
  ports:
  protocols: icmp ipv6-icmp
  forward: no
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
        rule priority="32767" reject
```

3. Add a rule that allows TCP traffic on port 80 for nm-shared zone, execute the following command:

   sudo firewall-cmd –permanent –add-port=80/tcp –zone=nm-shared

```
[root@server1 httpd]# sudo firewall-cmd --permanent --add-port=80/tcp --zone=nm-shared
success
[root@server1 httpd]#
```

4. Reload to apply permanent firewall rules to the running configuration.

   sudo firewall-cmd –reload

```
[root@server1 httpd]# sudo firewall-cmd --reload
success
```

5. List the firewall sets for the zone

sudo firewall-cmd --list-all --zone=nm-shared

```
[root@server1 httpd]# sudo firewall-cmd --list-all --zone=nm-shared
nm-shared (active)
  target: ACCEPT
  icmp-block-inversion: no
  interfaces: ens192
  sources:
  services: dhcp dns ssh
  ports: 80/tcp
  protocols: icmp ipv6-icmp
  forward: no
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
        rule priority="32767" reject
[root@server1 httpd]#
```

2.6 Add the ip addresses to be used for the project

2.6.1 Server

The /etc/hosts file is used for local DNS resolution, mapping IP addresses to hostnames. It bypasses DNS queries and directly associates a hostname (e.g., server1) with an IP address.

Edit /etc/hosts files

add the ip's to be used in the project

vim /etc/hosts

```
[root@server1 ~]# cat /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.50.10 server1.project.com
10.35.16.1 server1.project.com
10.35.17.1 server1.project.com
192.168.100.1 server1.project.com
```

nmcli con mod LAN1 +ipv4.addresses 10.35.16.1/24 +ipv4.addresses 10.35.17.1/24 +ipv4.addresses 192.168.100.1/24 +ipv4.addresses 10.50.1.1/24 +ipv4.addresses 10.51.1.1/24 +ipv4.addresses 10.52.1.1/24 +ipv4.addresses 10.53.1.1/24

nmcli connection down LAN1 ; nmcli con up LAN1

```
[root@server1 ~]#
nmcli con mod LAN1 +ipv4.addresses 10.35.16.1/24 +ipv4.addresses 10.35.17.1/24 +ipv4.addresses 192.168.100.1/24 +ipv4
.addresses 10.50.1.1/24 +ipv4.addresses 10.51.1.1/24 +ipv4.addresses 10.52.1.1/24 +ipv4.addresses 10.53.1.1/24
[root@server1 ~]# nmcli connection down LAN1 ; nmcli con up LAN1
Connection 'LAN1' successfully deactivated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/5)
Connection successfully activated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/6)
[root@server1 ~]#
```

nmcli

```
[root@server1 ~]# nmcli
ens160: connected to ens160
        "VMware VMXNET3"
        ethernet (vmxnet3), 00:0C:29:F6:C5:05, hw, mtu 1500
        ip4 default
        inet4 192.168.186.130/24
        route4 default via 192.168.186.2 metric 100
        route4 192.168.186.0/24 metric 100
        inet6 fe80::20c:29ff:fef6:c505/64
        route6 fe80::/64 metric 1024

ens192: connected to LAN1
        "VMware VMXNET3"
        ethernet (vmxnet3), 00:0C:29:F6:C5:0F, hw, mtu 1500
        inet4 10.53.1.1/24
        inet4 10.52.1.1/24
        inet4 10.51.1.1/24
        inet4 10.50.1.1/24
        inet4 192.168.100.1/24
        inet4 10.35.17.1/24
        inet4 10.35.16.1/24
        inet4 192.168.50.10/24
        route4 192.168.50.0/24 metric 101
        route4 10.35.16.0/24 metric 101
        route4 10.35.17.0/24 metric 101
        route4 192.168.100.0/24 metric 101
        route4 10.50.1.0/24 metric 101
        route4 10.51.1.0/24 metric 101
        route4 10.52.1.0/24 metric 101
        route4 10.53.1.0/24 metric 101
        route4 default via 192.168.50.1 metric 101
        inet6 fe80::3e70:81df:4f1a:15be/64
        route6 fe80::/64 metric 1024

lo: connected (externally) to lo
        "lo"
        loopback (unknown), 00:00:00:00:00:00, sw, mtu 65536
        inet4 127.0.0.1/8
        inet6 ::1/128
        route6 ::1/128 metric 256

DNS configuration:
        servers: 192.168.186.2
        domains: localdomain
        interface: ens160

        servers: 8.8.8.8
        interface: ens192

Use "nmcli device show" to get complete information about known devices and
"nmcli connection show" to get an overview on active connection profiles.

Consult nmcli(1) and nmcli-examples(7) manual pages for complete usage details.
[root@server1 ~]#
```

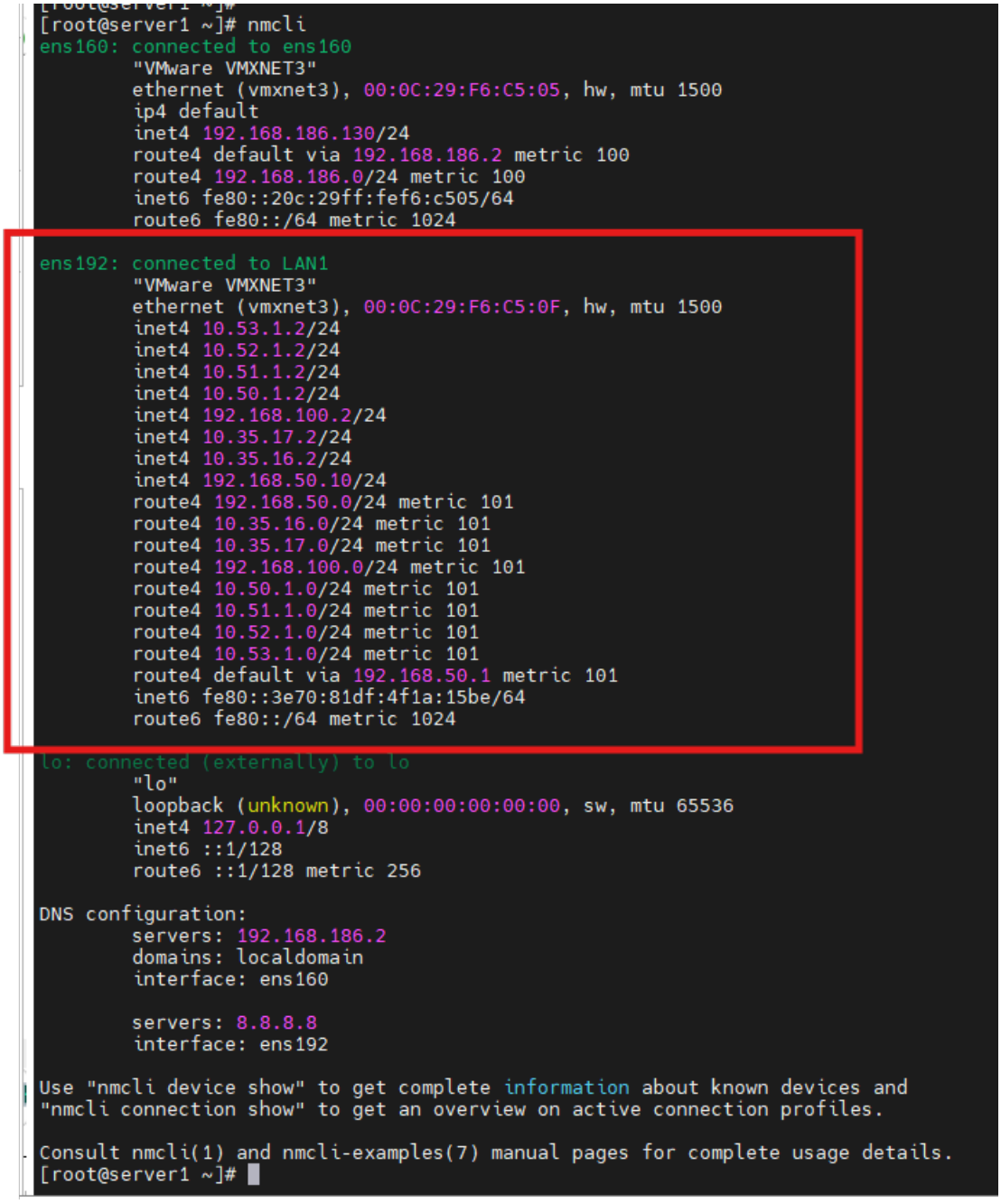
2.6.2 Client

Add needed Ip's to connection LAN1 in client Ubuntu

nmcli con mod LAN1 +ipv4.addresses 10.35.16.2/24 +ipv4.addresses 10.35.17.2/24 +ipv4.addresses 192.168.100.2/24 +ipv4.addresses 10.50.1.2/24 +ipv4.addresses 10.51.1.2/24 +ipv4.addresses 10.52.1.2/24 +ipv4.addresses 10.53.1.2/24

nmcli connection down LAN1 ; nmcli con up LAN1

List connection



3 TASK 1 – CREATING A WEBSITE

3.1 Requirements

1. Install and configure the Apache server to read web pages from the directory "/var/www/html_project1".
    - You must copy the httpd.conf file to httpd.conf.original.
    - Configure the server to be accessible via the IP address 192.168.50.10.
    - Ensure the httpd service is started and enabled to run at boot.
2. Create a new homepage named index.html in the/var/www/html_ project1 directory.
3. The page must include all of the following HTML tags (each used at least once):
   <html>, <head>, <title>, <body>, <p>, <hr>, <a href>, <img>,
, <b>, <i>, and <u>.
   - The page title must be Project Part I - Homepage
   - You must add a link to each web page of this project inside the index.html page.
   - Add a hyperlink to each web page created for this project inside the index.html page. Each link should open the corresponding web page of each question, allowing you to test its functionality.

3.2 Configure the Apache server

3.2.1 Create and setup the Document Root Directory

3.2.1.1 Create directory

The document root directory is where Apache looks for the files that will be served to website visitors. By default, Apache uses /var/www/html as the document root.

tree /var/www

```
[root@server1 ~]#
[root@server1 ~]# tree /var/www
/var/www
├── cgi-bin
└── html

2 directories, 0 files
[root@server1 ~]#
```

However, the user query specifies serving web pages from the /var/www/html_project1 directory. To create this directory, if it does not already exist, the following command should be executed:

sudo mkdir -p /var/www/html_project1

```
2 directories, 0 files
[root@server1 ~]# sudo mkdir -p /var/www/html_project1
```

This newly created directory will serve as the designated location for storing all the HTML files, images, scripts, and other assets that constitute the web project.

tree /var/www

```
[root@server1 ~]# tree /var/www/
/var/www/
├── cgi-bin
├── html
└── html_project1

3 directories, 0 files
[root@server1 ~]#
```

3.2.1.2 Modify permissions and ownership

1. Subdirectories and files stored in this directory should be readable by everyone using the privilege:

   chmod –R 755 /var/www/html_project1

```
[root@server1 html_project1]# chmod -R 755 /var/www/html_project1
```

2. Verify permissions of file

   ls -lqrtha /var/www/

```
[root@server1 html_project1]# ls -lqrtha /var/www/
total 4.0K
drwxr-xr-x.  2 root root    6 Jan 21 16:23 html
drwxr-xr-x.  2 root root    6 Jan 21 16:23 cgi-bin
drwxr-xr-x. 21 root root 4.0K Apr 16 00:50 ..
drwxr-xr-x.  5 root root   54 Apr 16 14:31 .
drwxr-xr-x.  2 root root   24 Apr 17 01:08 html_project1
```

3. Change file ownership

   sudo chown -R apache:apache /var/www/html_project1

```
[root@server1 html_project1]# sudo chown -R apache:apache /var/www/html_project1
```

4. Verify ownership of file

   ls -lrhtqa /var/www/

```
[root@server1 html_project1]# ls -lrhtqa /var/www/
total 4.0K
drwxr-xr-x.  2 root   root      6 Jan 21 16:23 html
drwxr-xr-x.  2 root   root      6 Jan 21 16:23 cgi-bin
drwxr-xr-x. 21 root   root   4.0K Apr 16 00:50 ..
drwxr-xr-x.  5 root   root     54 Apr 16 14:31 .
drwxr-xr-x.  2 apache apache   43 Apr 17 01:44 html_project1
```

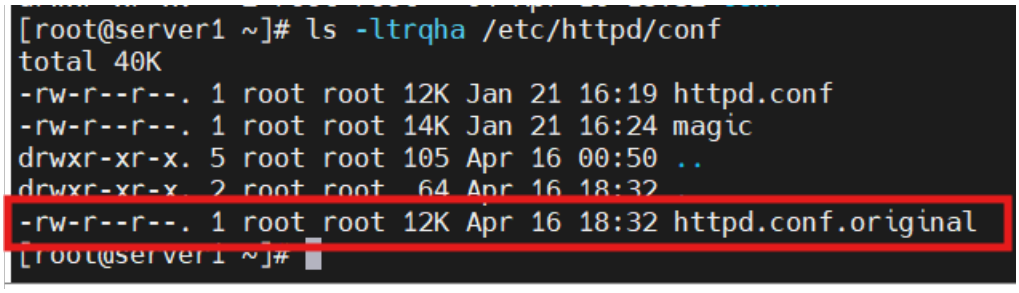
3.2.2 Global configuration file (httpd.conf)

1. Copy the httpd.conf file to httpd.conf.original

   This command copies the contents of the original httpd.conf file to a new file named httpd.conf.original within the same directory. If any unintended changes are made or if the new configuration causes problems we can rollback to original configuration.

   cp /etc/httpd/conf/httpd.conf /etc/httpd/conf/httpd.conf.original

2. Verify file is created
   ls -lqrtha /etc/httpd/conf



3. Modify the Listen directive in the httpd.conf file. Change the default listening address from all interfaces on port 80 to the specific IP address 192.168.50.10 on port 80.

a) Modify the file /etc/httpd/conf/httpd.conf according to the following table:

Current Configuration	**Updated Configuration**
ServerName ServerName	192.168.50.10 server1.project.com
DocumentRoot "/var/www/html"	DocumentRoot "/var/www/html_project1"
<Directory "/var/www/html">	<Directory "/var/www/html_project1"> Options Indexes FollowSymLinks AllowOverride All Require all granted </Directory>

Open the file for edition and update the configuration according to table above

vim /etc/httpd/conf/httpd.conf

```
ServerRoot "/etc/httpd"
Listen 80
Include conf.modules.d/*.conf

User apache
Group apache

#ServerAdmin root@localhost
ServerName 192.168.50.10
ServerName server1.project.com

<Directory />
    AllowOverride none
    Require all denied
</Directory>

DocumentRoot "/var/www/html_project1"

<Directory "/var/www">
    AllowOverride None
    Require all granted
</Directory>

<Directory "/var/www/html_project1">
    Options Indexes FollowSymLinks
    AllowOverride All
    Require all granted
</Directory>
```

b) Ensure the httpd config syntax is correct

httpd -t

```
[root@server1 html_project1]# httpd -t
Syntax OK
[root@server1 html_project1]#
```

Expected syntax is ok , if any error go back and correct it.

c) Reload Apache:

systemctl reload httpd

```
[root@server1 ~]# systemctl reload httpd
[root@server1 ~]#
```

3.2.3 Index file for Root directory

1. Create a new file named **index.html** in the**/var/www/html_ project1** directory.

touch /var/www/html_project1/index.html

```
[root@server1 html_project1]# touch /var/www/html_project1/index.html
```

2. As per requirements the page must include all of the following **HTML tags** (each used at least once): <html>, <head>, <title>, <body>, <p>, <hr>, <a href>, <img>,
, , , and <u>.

Tag	Description
<html>	The root element that defines the entire HTML document. It wraps all the content of the page.
<head>	Contains metadata about the document, such as the title, links to stylesheets, and scripts.
<title>	Specifies the title of the web page, shown on the browser tab and in search engine results.
<body>	Encloses the content visible to the user on the web page, like text, images, and links.
<p>	Defines a paragraph of text.
<hr>	Creates a horizontal line, often used to visually separate content sections.
<a href>	Creates a hyperlink to another web page or resource. Example: Link text.
<img>	Displays an image on the page. The src attribute specifies the image source.
 	Inserts a line break in the text, similar to pressing "Enter" in a document.
	Makes text bold. Example: Bold text.
	Makes text italic. Example: Italic text.
<u>	Underlines text. Example: <u>Underlined text</u>.

3. Create index file make sure to include all the HTML requirement mentioned above.

vim /var/www/html_project1/index.html

```
<!DOCTYPE html>
<html>
<head>
 <title>Project Part I - Homepage</title>
</head>
<body>
    <p><b><i><u>Welcome to Project Part I</u></i></b></p>
    <hr>
    <a href="task2.html">Task 2 - Secure Directories</a><br>
    <a href="project1.html">Task 3 - Project 1</a><br>
    <a href="project2.html">Task 3 - Project 2</a><br>
    <a href="project3.html">Task 3 - Project 3</a><br>
    <a href="project4.html">Task 3 - Project 4</a><br>
    <a href="vendors.html">Task 4 - Vendors Website</a><br>
    <a href="accountants.html">Task 4 - Accountants
Website</a><br>
    <a href="programmers.html">Task 4 - Programmers
Website</a><br>
    <a href="administrators.html">Task 4 - Administrators
Website</a><br>
    <img src="example.jpg" alt="Example Image"><br>
</body>
</html>
```

```
<!DOCTYPE html>
<html>
<head>
    <title>Project Part I - Homepage</title>
</head>
<body>
    <p><b><i><u>Welcome to Project Part I</u></i></b></p>
    <hr>
    <a href="task2.html">Task 2 - Secure Directories</a><br>
    <a href="project1.html">Task 3 - Project 1</a><br>
    <a href="project2.html">Task 3 - Project 2</a><br>
    <a href="project3.html">Task 3 - Project 3</a><br>
    <a href="project4.html">Task 3 - Project 4</a><br>
    <a href="vendors.html">Task 4 - Vendors Website</a><br>
    <a href="accountants.html">Task 4 - Accountants Website</a><br>
    <a href="programmers.html">Task 4 - Programmers Website</a><br>
    <a href="administrators.html">Task 4 - Administrators Website</a><br>
    <img src="example.jpg" alt="Example Image"><br>
</body>
</html>
~
~
```

4. Verify file is created

   ls -lqrtha /var/www/html_project1

```
[root@server1 html_project1]# ls -lqrtha /var/www/html_project1
total 4.0K
drwxr-xr-x. 5 root root  54 Apr 16 14:31 ..
-rw-r--r--. 1 root root 776 Apr 17 01:08 index.html
drwxr-xr-x. 2 root root  24 Apr 17 01:08 .
```

5. Create image file
   This command retrieves an image file from the specified URL and saves it as */var/www/html_project1/example.jpg* on your server.

   wget https://picsum.photos/600/400 -O /var/www/html_project1/images/example.jpg

```
[root@server1 html_project1]# wget https://picsum.photos/600/400 -O /var/www/html_project1/example.jpg
--2025-04-17 12:09:43--  https://picsum.photos/600/400
Resolving picsum.photos (picsum.photos)... 104.26.4.30, 172.67.74.163, 104.26.5.30, ...
Connecting to picsum.photos (picsum.photos)|104.26.4.30|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://fastly.picsum.photos/id/62/600/400.jpg?hmac=GxWI7cJivRMdEntglLHv2LPQwFq34DCH700U-BBjQ2Y [following]
--2025-04-17 12:09:44--  https://fastly.picsum.photos/id/62/600/400.jpg?hmac=GxWI7cJivRMdEntglLHv2LPQwFq34DCH700U-BBjQ2Y
Resolving fastly.picsum.photos (fastly.picsum.photos)... 151.101.137.91, 2a04:4e42:20::347
Connecting to fastly.picsum.photos (fastly.picsum.photos)|151.101.137.91|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 26250 (26K) [image/jpeg]
Saving to: '/var/www/html_project1/example.jpg'

/var/www/html_project1/example.jpg       100%[=============================================================================================>]  25.63K  --.-KB/s    in 0.001s

2025-04-17 12:09:44 (30.7 MB/s) - '/var/www/html_project1/example.jpg' saved [26250/26250]

[root@server1 html_project1]#
```

6. Verify image file is created

   ls -lqrtha /var/www/html_project1/example.jpg

```
[root@server1 html_project1]# ls -lqrtha /var/www/html_project1/example.jpg
-rw-r--r--. 1 root root 26K Apr 17 12:09 /var/www/html_project1/example.jpg
[root@server1 html_project1]#
```

3.2.4 Start and enable Apache

Ensure the httpd service is started and enabled to run at boot.

systemctl start httpd

systemctl enable httpd

```
[root@server1 html_project1]# systemctl start httpd
[root@server1 html_project1]# systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
[root@server1 html_project1]#
```

Verify status

systemctl status httpd

```
[root@server1 html_project1]# systemctl status httpd
● httpd.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
     Active: active (running) since Wed 2025-04-16 10:43:44 EDT; 14h ago
       Docs: man:httpd.service(8)
   Main PID: 78466 (httpd)
     Status: "Total requests: 12; Idle/Busy workers 100/0;Requests/sec: 0.000224; Bytes served/sec:   0 B/sec"
      Tasks: 230 (limit: 22829)
     Memory: 45.7M
        CPU: 55.480s
     CGroup: /system.slice/httpd.service
             ├─78466 /usr/sbin/httpd -DFOREGROUND
             ├─78467 /usr/sbin/httpd -DFOREGROUND
             ├─78468 /usr/sbin/httpd -DFOREGROUND
             ├─78499 /usr/sbin/httpd -DFOREGROUND
             ├─78523 /usr/sbin/httpd -DFOREGROUND
             └─80044 /usr/sbin/httpd -DFOREGROUND

Apr 16 10:43:28 server1 systemd[1]: Starting The Apache HTTP Server...
Apr 16 10:43:39 server1 httpd[78466]: AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using fe80::20c:29ff:fef6:c505%ens160. S>
Apr 16 10:43:44 server1 systemd[1]: Started The Apache HTTP Server.
Apr 16 10:43:44 server1 httpd[78466]: Server configured, listening on: port 80
lines 1-21/21 (END)
```

3.3 Verify TASK 1

From a browser on the AlmaLinux VM:

You should now see your custom content, not the AlmaLinux test page.

Note Links still do not work is just the index.

Activities Firefox

AlmaLinux OS - Forever-F × Project Part I - Homepage × Project Part I - Homepage ×

192.168.50.10

AlmaLinux Documentation Blog Bug tracker GitHub organization

Welcome to Project Part I

Task 2 - Secure Directories
Task 3 - Project 1
Task 3 - Project 2
Task 3 - Project 3
Task 3 - Project 4
Task 4 - Vendors Website
Task 4 - Accountants Website
Task 4 - Programmers Website
Task 4 - Administrators Website

4 TASK 2 – AUTHENTICATION

4.1 Requirements

4.2 Modify main html

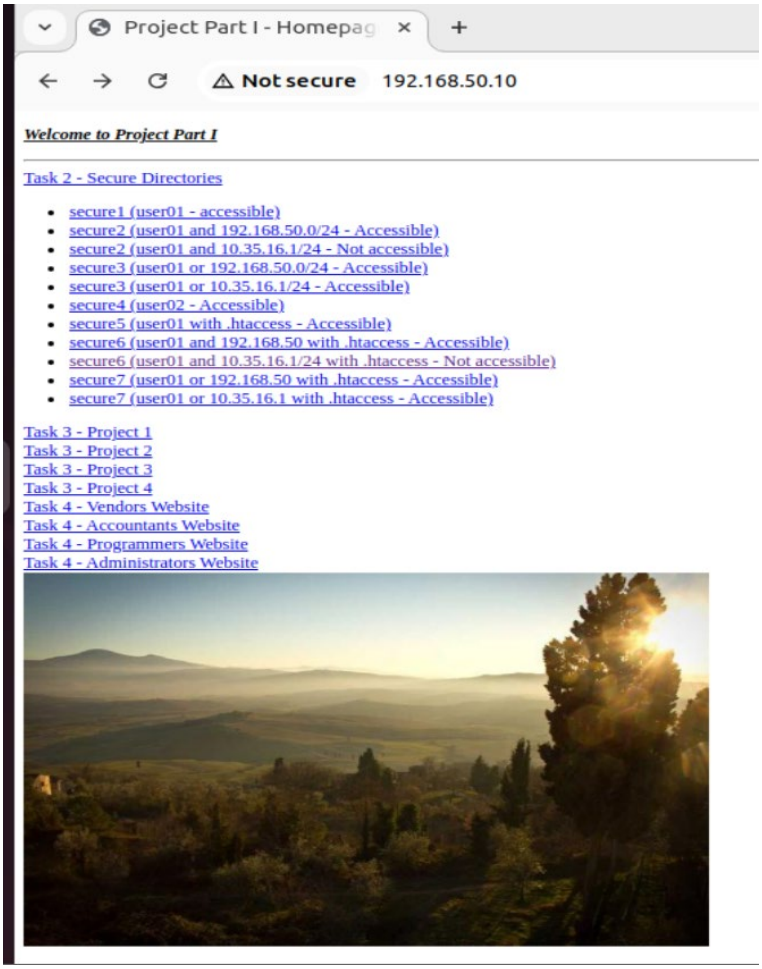
Modify main html file to include Task 2 menu

Open the file for edition and include the lines in the box.

vi /var/www/html_project1/index.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Project Part I - Homepage</title>
</head>
<body>
    <p><b><i><u>Welcome to Project Part I</u></i></b></p>
    <hr>

    <!-- Main Menu -->
    <a href="task2.html">Task 2 - Secure Directories</a>
    <ul>
        <!-- Sub-Menu for Task 2 -->
        <li><a href="http://192.168.50.10/secure1/index.html">secure1 (user01 -
accessible)</a></li>
        <li><a href="http://192.168.50.10/secure2/index.html">secure2 (user01 and 192.168.50.0/24
- Accessible)</a></li>
        <li><a href="http://10.35.16.1/secure2/index.html">secure2 (user01 and 10.35.16.1/24 -
Not accessible)</a></li>
        <li><a href="http://192.168.50.10/secure3/index.html">secure3 (user01 or 192.168.50.0/24
- Accessible)</a></li>
        <li><a href="http://10.35.16.1/secure3/index.html">secure3 (user01 or 10.35.16.1/24 - Not
accessible)</a></li>
        <li><a href="http://192.168.50.10/secure4/index.html">secure4 (user02 -
Accessible)</a></li>
        <li><a href="http://192.168.50.10/secure5/index.html">secure5 (user01 with .htaccess -
Accessible)</a></li>
        <li><a href="http://192.168.50.10/secure6/index.html">secure6 (user01 and 192.168.50 with
.htaccess - Accessible)</a></li>
        <li><a href="http://10.35.16.1/secure6/index.html">secure6 (user01 and 10.35.16.1/24 with
.htaccess - Not accessible)</a></li>
        <li><a href="http://192.168.50.10/secure7/index.html">secure7 (user01 or 192.168.50 with
.htaccess - Accessible)</a></li>
        <li><a href="http://10.35.16.1/secure7/index.html">secure7 (user01 or 10.35.16.1 with
.htaccess - Accessible)</a></li>
    </ul>

    <a href="project1.html">Task 3 - Project 1</a><br>
    <a href="project2.html">Task 3 - Project 2</a><br>
    <a href="project3.html">Task 3 - Project 3</a><br>
    <a href="project4.html">Task 3 - Project 4</a><br>
    <a href="vendors.html">Task 4 - Vendors Website</a><br>
    <a href="accountants.html">Task 4 - Accountants Website</a><br>
    <a href="programmers.html">Task 4 - Programmers Website</a><br>
    <a href="administrators.html">Task 4 - Administrators Website</a><br>
    <img src="example.jpg" alt="Example Image"><br>
</body>
</html>
```

Verify web page display (links still not working)



4.3 Users

1. Create users

   Create the user authentication file:

   htpasswd -c /etc/httpd/.htpasswd user01

   # Enter password: secret

   htpasswd /etc/httpd/.htpasswd user02

   # Enter password: secret

```
[root@server1 html_project1]# sudo htpasswd -c /etc/httpd/.htpasswd user01
New password:
Re-type new password:
Adding password for user user01
[root@server1 html_project1]# sudo htpasswd /etc/httpd/.htpasswd user02
New password:
Re-type new password:
Adding password for user user02
[root@server1 html_project1]#
```


2. Check the authentication file and verify users were created

   cat /etc/httpd/.htpasswd

```
[root@server1 html_project1]# cat /etc/httpd/.htpasswd
user01:$apr1$1x/OzTUQ$E6yPFT1JvnLBnkmdcBm141
user02:$apr1$uaPe6pkN$Tx9vAy4jOVGuvbIk43uG.1
[root@server1 html_project1]#
```

3. Manually Verify the Users Credentials
   a) Verify user01

   htpasswd -v /etc/httpd/.htpasswd user01

   When prompted to enter the password for user01 use *secret*

```
[root@server1 html_project1]# htpasswd -v /etc/httpd/.htpasswd user01
Enter password:
Password for user user01 correct.
```

   b) Verify user02

   htpasswd -v /etc/httpd/.htpasswd user02

   When prompted to enter the password for user02 use *secret*

```
[root@server1 html_project1]# htpasswd -v /etc/httpd/.htpasswd user02
Enter password:
Password for user user02 correct.
[root@server1 html_project1]#
```

4.4 Secure1

Create the directory and configuration to fulfill the following requirements:

Create secure1 directory and configure Apache using the <Directory> directive so that only the user user01 with the password “secret” can access it from any subnet. No other users should have access.

4.4.1 Directories

1. Create directory

   mkdir /var/www/html_project1/secure1

```
[root@server1 html_project1]# mkdir /var/www/html_project1/secure1
```

2. Verify directory is created

   ls -lqrtha /var/www/html_project1/

```
[root@server1 html_project1]# ls -lqrtha /var/www/html_project1/
total 32K
drwxr-xr-x. 5 root   root    54 Apr 16 14:31 ..
-rw-r--r--. 1 apache apache 776 Apr 17 01:08 index.html
-rw-r--r--. 1 root   root   26K Apr 17 12:09 example.jpg
drwxr-xr-x. 2 root   root     6 Apr 17 12:49 secure1
drwxr-xr-x. 3 apache apache  58 Apr 17 12:50 .
[root@server1 html_project1]#
```

4.4.2 Configuration

4.4.2.1 File httpd.conf

1. Modify configuration file */etc/httpd/conf/httpd.conf* to fulfill requirements

   Create a secure1 directory and configure Apache using the <Directory> directive so that only the user user01 with the password “secret” can access it from any subnet. No other users should have access.

   a) Add the following lines to create a new Directory block

   This block should be added after the existing <Directory> block for /var/www/html_project1 to maintain logical order and avoid overriding other directory settings.

```
<Directory "/var/www/html_project1/secure1">
    AuthType Basic
    AuthName "Restricted Access - secure1"
    AuthUserFile /etc/httpd/.htpasswd
    Require user user01
</Directory>
```

   b) Open the file for edition

   vim /etc/httpd/conf/httpd.conf

```
<Directory "/var/www">
    AllowOverride None
    Require all granted
</Directory>

<Directory "/var/www/html_project1">
    Options Indexes FollowSymLinks
    AllowOverride All
    Require all granted
</Directory>

<Directory "/var/www/html_project1/secure1">
    AuthType Basic
    AuthName "Restricted Access - secure1"
    AuthUserFile /etc/httpd/.htpasswd
    Require user user01
</Directory>
```

c) Verify the syntax of configuration file after changes

httpd -t

```
[root@server1 conf]# httpd -t
Syntax OK
[root@server1 conf]#
```

d) Reload Apache:

systemctl reload httpd

```
[root@server1 ~]# systemctl reload httpd
[root@server1 ~]#
```

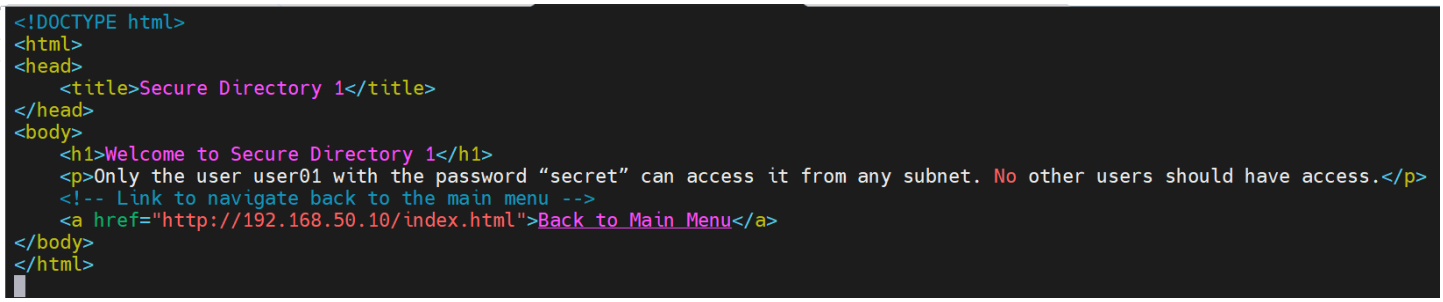
4.4.2.2 File index.html for secure1

1. Create index for secure1

vim /var/www/html_project1/secure1/index.html

Add the following lines

```
<!DOCTYPE html>
<html>
<head>
    <title>Secure Directory 1</title>
</head>
<body>
    <h1>Welcome to Secure Directory 1</h1>
    <p>Only the user user01 with the password “secret” can access it from any subnet. No
other users should have access.</p>
    <!-- Link to navigate back to the main menu -->
    <a href="http://192.168.50.10/index.html">Back to Main Menu</a>
</body>
</html>
```



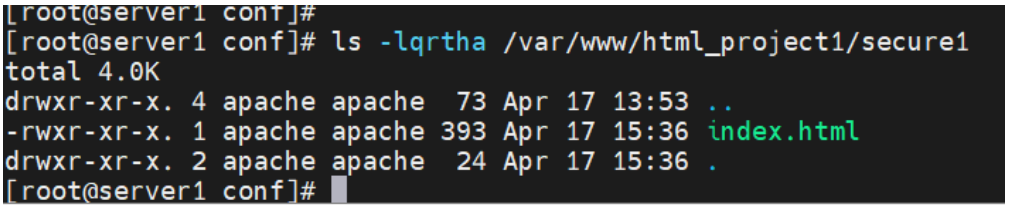
2. Change file permissions and ownership

   chown -R apache:apache /var/www/html_project1/secure1

   chmod -R 755 /var/www/html_project1/secure1

3. Verify changes are applied

   ls -lqrtha /var/www/html_project1/secure1



4.4.3 Test secure 1

Test the requirement - Only the user user01 with the password “secret” can access it from any subnet. No other users should have access.

4.4.3.1 Browser test

Test is done from ubuntu using chrome browser

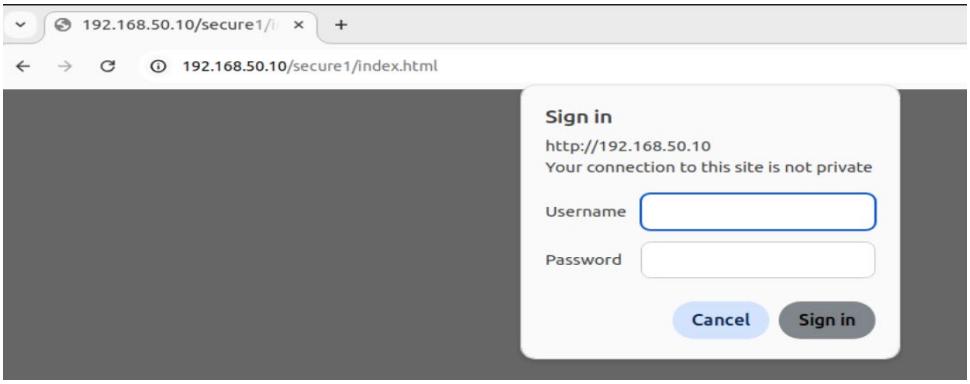
4.4.3.1.1 Test Only the user user01 can access from any subnet.

1. Open chrome
2. Set ip 192.168.50.10

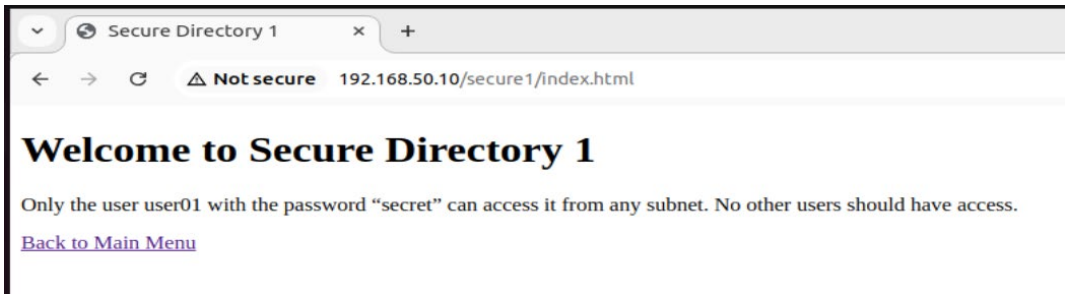
   See main menu page is displayed

3. Select **securel (user01 - accessible)** link and double click

   You are prompted for user/password write user01/secret



4. The following page appears



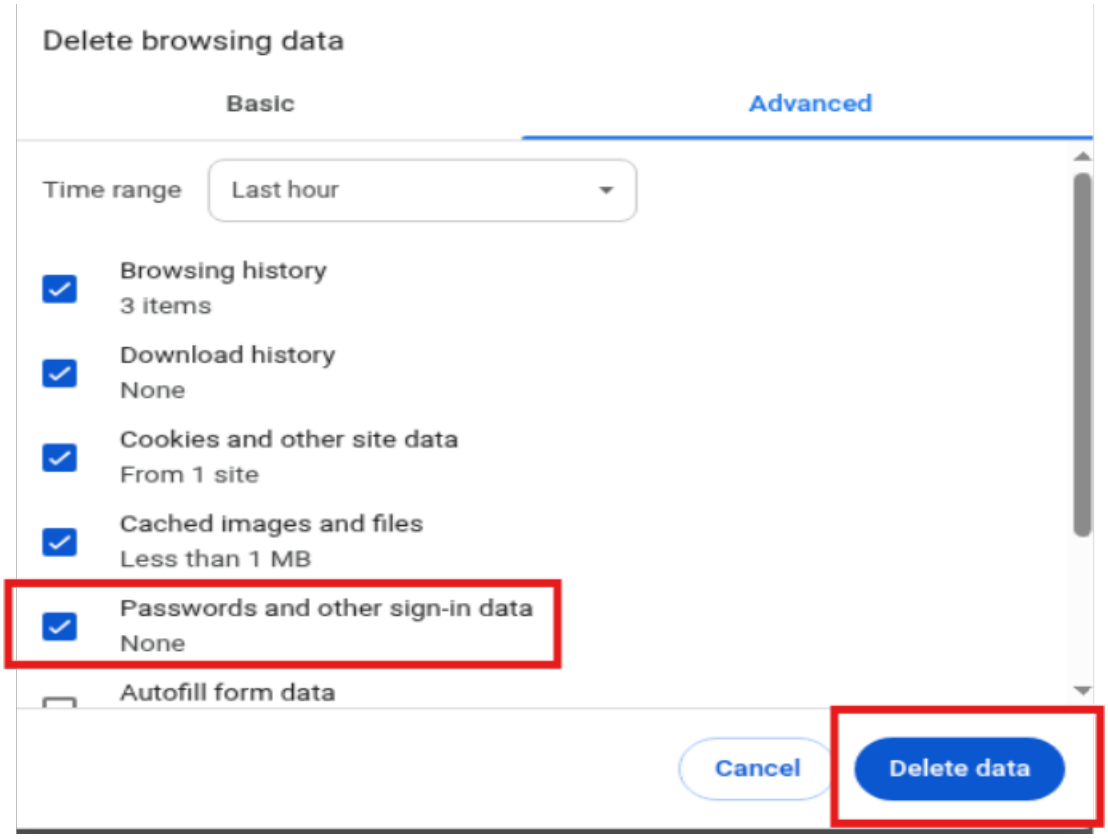
4.4.3.1.2 Test No other users should have access.

1. Clear cache

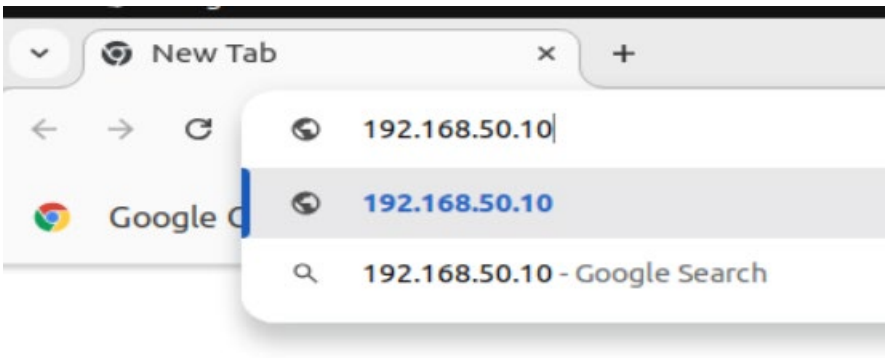
   Go back to main menu and clear cache (ctrl-shift-del)

   A menu appears select Advanced and make sure all choices in picture below are selected.

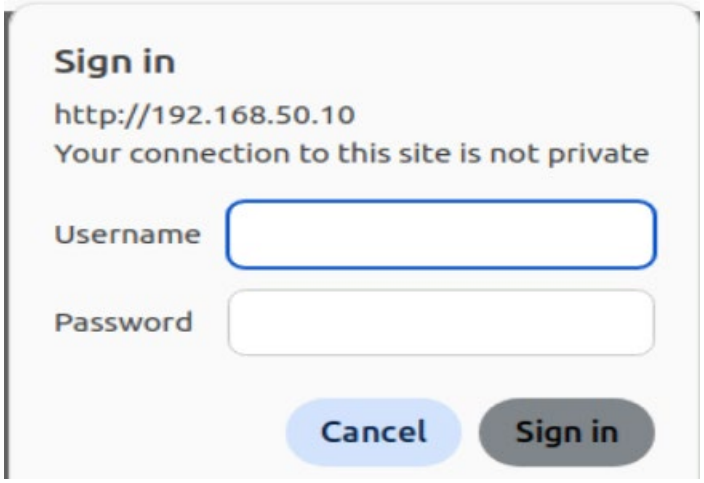
   Press delete data.



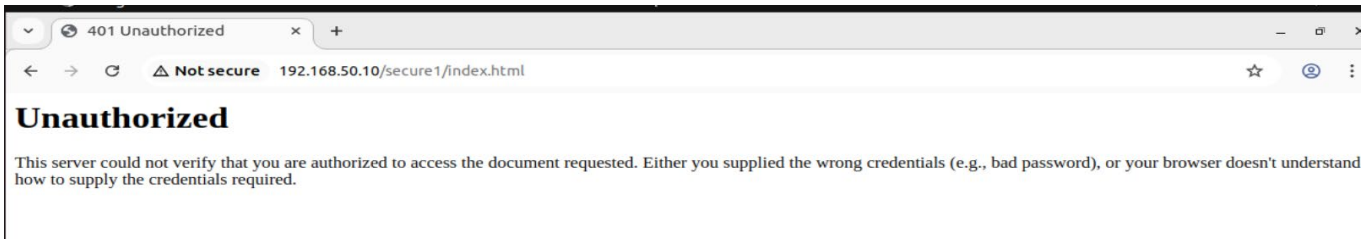
2. Open a fresh page and write 192.168.50.10



3. As instructed above , in main page : Select Task2-Secure Directories/Secure Directory 1
4. When prompted for user / password give **user02/secret**



5. See you are asked for user/password again (even when right user /password was given) Select cancel and see the following is displayed



4.5 secure2

Create the directory and configuration to fulfill the following requirements:

Create a ***secure2*** directory and configure Apache so that only user01 can access it when connecting from the subnet 192.168.50.0/24.

4.5.1 Directories

1. Create directory

   mkdir /var/www/html_project1/secure2

   ```
   [root@server1 conf]#  mkdir /var/www/html_project1/secure2
   ```

2. Verify directory is created

   ls -lqrtha /var/www/html_project1/

```
[root@server1 html_project1]# ls -lqrtha /var/www/html_project1/
total 32K
drwxr-xr-x. 5 root   root     54 Apr 16 14:31 ..
-rw-r--r--. 1 root   root    26K Apr 17 12:09 example.jpg
drwxr-xr-x. 2 root   root      6 Apr 17 13:53 secure2
drwxr-xr-x. 2 apache apache   24 Apr 17 16:27 secure1
-rw-r--r--. 1 apache apache 3.3K Apr 17 22:29 index.html
drwxr-xr-x. 4 apache apache   73 Apr 18 00:53 .
[root@server1 html_project1]#
```

4.5.2 Configuration

4.5.2.1 File httpd.conf

Modify configuration file */etc/httpd/conf/httpd.conf* to fulfill requirements

Create a **secure2** directory and configure Apache so that only **user01** can access it when connecting from the subnet **192.168.50.0/24**.

a) Open the file for edition

vim /etc/httpd/conf/httpd.conf

b) Add the following lines to create a new Directory block

```
<Directory "/var/www/html_project1/secure2">
  AuthType Basic
  AuthName "Restricted Access - secure2"
  AuthUserFile /etc/httpd/.htpasswd
  <RequireAll>
    Require user user01
    Require ip 192.168.50.0/24
  </RequireAll>
</Directory>
```

```
<Directory "/var/www/html_project1/secure1">
    AuthType Basic
    AuthName "Restricted Access - secure1"
    AuthUserFile /etc/httpd/.htpasswd
    Require user user01
</Directory>

<Directory "/var/www/html_project1/secure2">
    AuthType Basic
    AuthName "Restricted Access - secure2"
    AuthUserFile /etc/httpd/.htpasswd
    <RequireAll>
        Require user user01
        Require ip 192.168.50.0/24
    </RequireAll>
</Directory>
```

c) Verify the syntax of configuration file after changes

httpd -t

```
[root@server1 conf]# httpd -t
Syntax OK
[root@server1 conf]#
```

d) Reload Apache:

systemctl reload httpd

```
[root@server1 ~]# systemctl reload httpd
[root@server1 ~]#
```

4.5.2.2 File index.html for secure2

1. Create index for secure2

vim /var/www/html_project1/secure2/index.html

```
[root@server1 html_project1]#
[root@server1 html_project1]# vim  /var/www/html_project1/secure2/index.html
[root@server1 html_project1]#
```

Add the following lines

```
<!DOCTYPE html>
<html>
<head>
  <title>Secure Directory 2</title>
</head>
<body>
  <h1>Welcome to Secure Directory 2</h1>
  <p>only user01 can access it when connecting from the subnet 192.168.50.0/24.</p>
  <!-- Link to navigate back to the main menu -->
  <a href="http://192.168.50.10/index.html">Back to Main Menu</a>
</body>
</html>
```

```
<!DOCTYPE html>
<html>
<head>
    <title>Secure Directory 2</title>
</head>
<body>
    <h1>Welcome to Secure Directory 2</h1>
    <p>only user01 can access it when connecting from the subnet 192.168.50.0/24.</p>
    <!-- Link to navigate back to the main menu -->
    <a href="http://192.168.50.10/index.html">Back to Main Menu</a>
</body>
</html>
```

2. Change file permissions and ownership

   chown -R apache:apache /var/www/html_project1/secure2

   chmod -R 755 /var/www/html_project1/secure2

```
[root@server1 html_project1]# chown -R apache:apache /var/www/html_project1/secure2
[root@server1 html_project1]# chmod -R 755 /var/www/html_project1/secure2
[root@server1 html_project1]#
```

3. Verify changes are applied

   ls -lqrtha /var/www/html_project1/secure2

```
[root@server1 html_project1]# ls -lqrtha /var/www/html_project1/secure2
total 4.0K
drwxr-xr-x. 4 apache apache  73 Apr 18 00:53 ..
-rwxr-xr-x. 1 apache apache 249 Apr 18 03:12 index.html
drwxr-xr-x. 2 apache apache  24 Apr 18 03:12 .
[root@server1 html_project1]#
```

4.5.3 Test secure 2

4.5.3.1 Browser test

Test is done from ubuntu using chrome browser

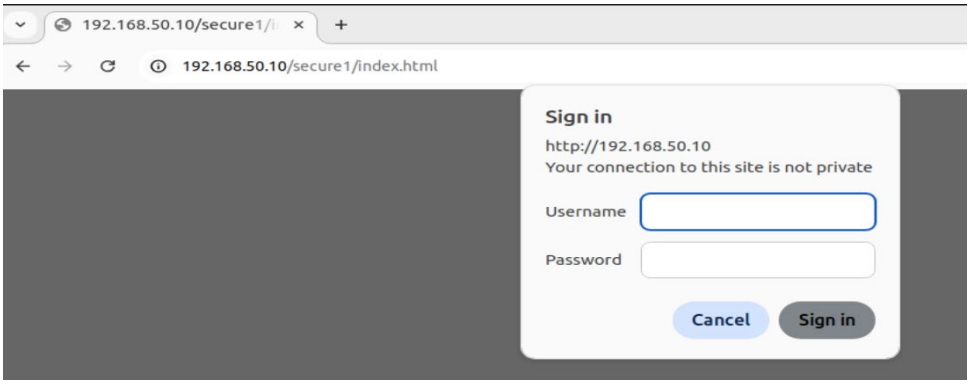
4.5.3.1.1 secure2: (user01 and 192.168.50.0/24 - accessible)

1. Open chrome
2. Set ip 192.168.50.10

   Main page is displayed, select secure2 (first choice):

   secure2: (user01 and 192.168.50.0/24 - accessible)

3. When prompted for user/password write user01/secret



4. The following page appears

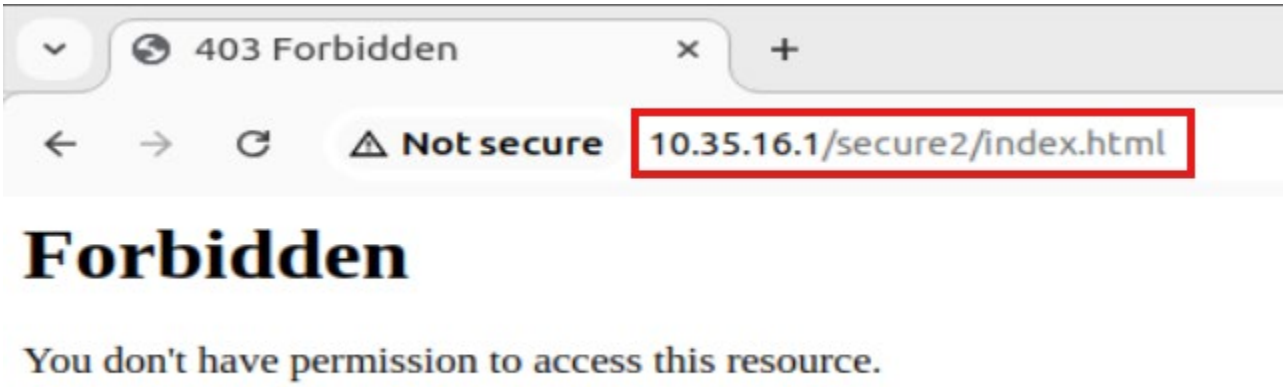
4.5.3.1.2 secure2: (user01 and 10.35.16.1/24 - Not accessible)

1. Open chrome
2. Set ip 192.168.50.10

   See main page is displayed, select secure2 (second choice)

   Select secure2: (user01 and 10.35.16.1/24 - Not accessible)

3. Forbidden appears note ip 10.35.16.1



4.6 Secure3

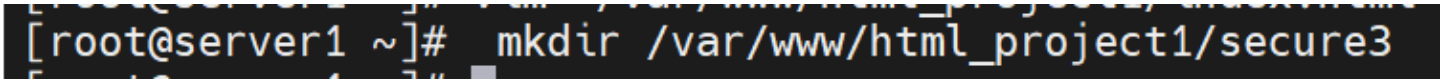
Create the directory and configuration to fulfill the following requirements:

Create a **secure3** directory and configure Apache so that either user01 or any user from the 192.168.50.0/24 subnet can access it.

4.6.1 Directories

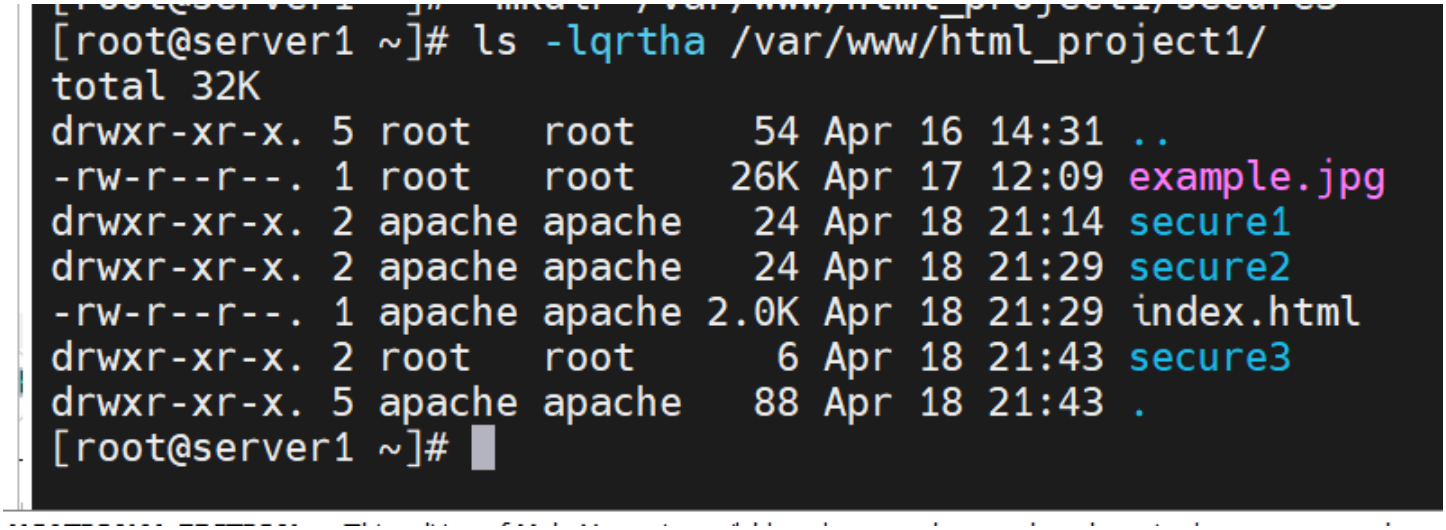
1. Create directory

   mkdir /var/www/html_project1/secure3



2. Verify directory is created

   ls -lqrtha /var/www/html_project1/



3. Modify httpd configuration

vim /etc/httpd/conf/httpd.conf

```
<Directory "/var/www/html_project1/secure3">
  AuthType Basic
  AuthName "Restricted Access - secure3"
  AuthUserFile /etc/httpd/.htpasswd
  <RequireAny>
    Require user user01
    Require ip 192.168.50.0/24
  </RequireAny>
</Directory>
```

```
</Directory>

<Directory "/var/www/html_project1/secure2">
    AuthType Basic
    AuthName "Restricted Access - secure2"
    AuthUserFile /etc/httpd/.htpasswd
    <RequireAll>
        Require user user01
        Require ip 192.168.50.0/24
    </RequireAll>
</Directory>

<Directory "/var/www/html_project1/secure3">
    AuthType Basic
    AuthName "Restricted Access - secure3"
    AuthUserFile /etc/httpd/.htpasswd
    <RequireAny>
        Require user user01
        Require ip 192.168.50.0/24
    </RequireAny>
</Directory>
```

4. Verify the syntax of configuration file after changes

httpd -t

```
[root@server1 conf]# httpd -t
Syntax OK
[root@server1 conf]#
```

4.6.2 Configuration

4.6.2.1 File httpd.conf

Modify configuration file */etc/httpd/conf/httpd.conf* to fulfill requirements

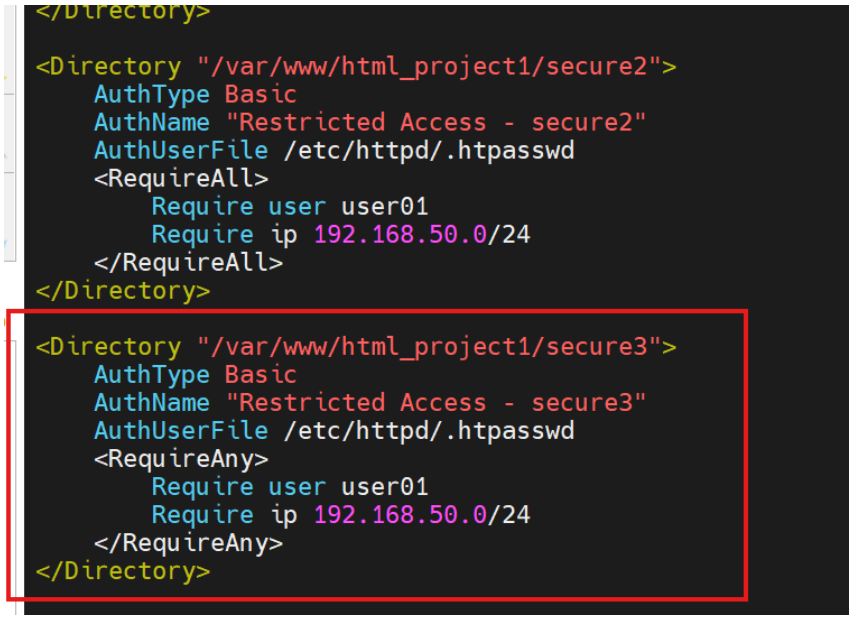
Create a **secure3** directory and configure Apache so that either **user01** or **any user** from the **192.168.50.0/24** subnet can access it.

e) Open the file for edition

vim /etc/httpd/conf/httpd.conf

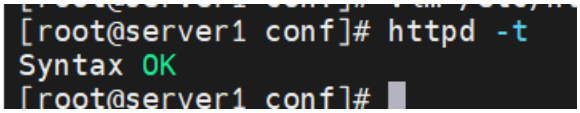
f) Add the following lines to create a new Directory block

```
<Directory "/var/www/html_project1/secure3">
  AuthType Basic
  AuthName "Restricted Access - secure3"
  AuthUserFile /etc/httpd/.htpasswd
  <RequireAny>
    Require user user01
    Require ip 192.168.50.0/24
  </RequireAny>
</Directory>
```



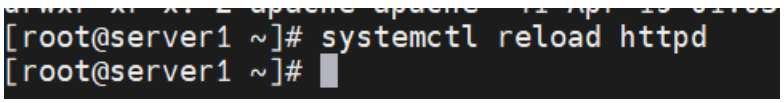
g) Verify the syntax of configuration file after changes

httpd -t



h) Reload Apache:

systemctl reload httpd



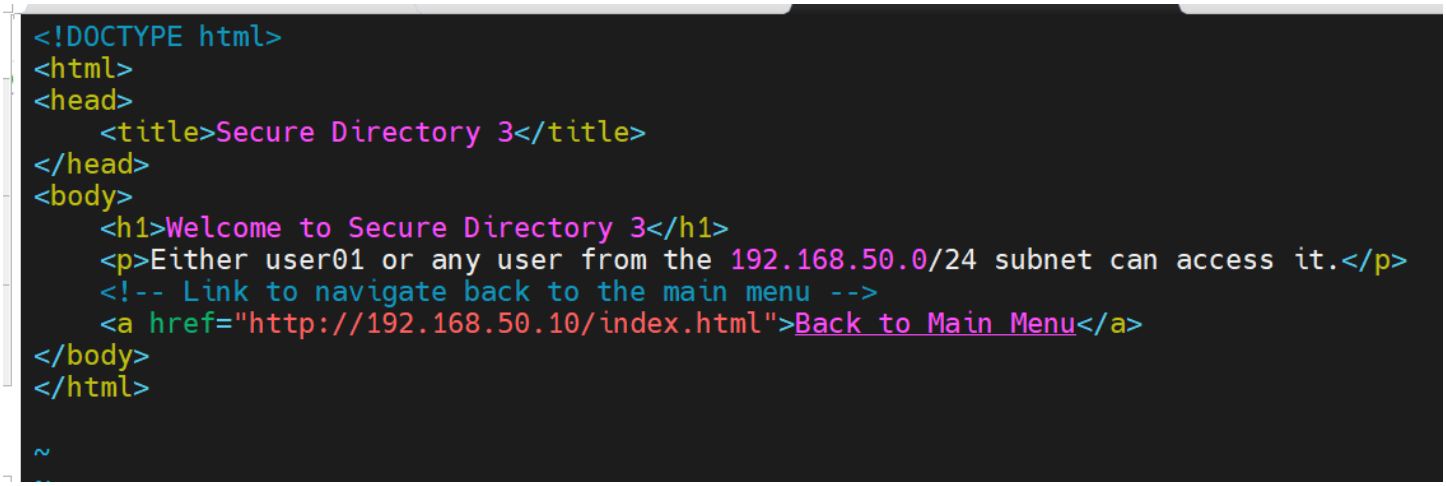
4.6.2.2 File index.html for secure3

1. Create index for secure3

vim /var/www/html_project1/secure3/index.html

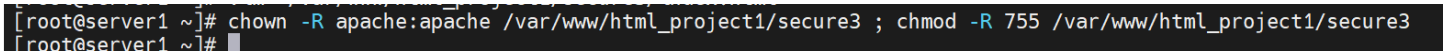
Add the following lines

```
<!DOCTYPE html>
<html>
<head>
  <title>Secure Directory 3</title>
</head>
<body>
  <h1>Welcome to Secure Directory 3</h1>
  <p>Either user01 or any user from the 192.168.50.0/24 subnet can access it.</p>
  <!-- Link to navigate back to the main menu -->
  <a href="http://192.168.50.10/index.html">Back to Main Menu</a>
</body>
</html>
```



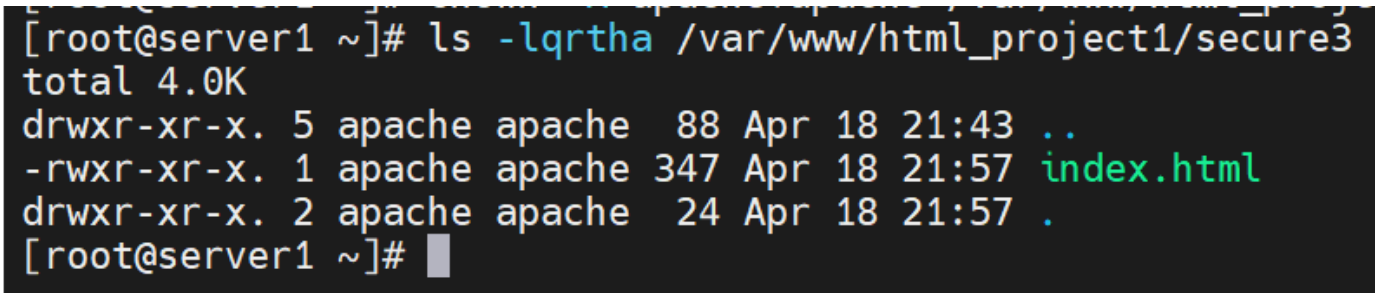
2. Change file permissions and ownership

   chown -R apache:apache /var/www/html_project1/secure3 ; chmod -R 755 /var/www/html_project1/secure3



3. Verify changes are applied

   ls -lqrtha /var/www/html_project1/secure3



4.6.3 Test secure 3

4.6.3.1 Browser test

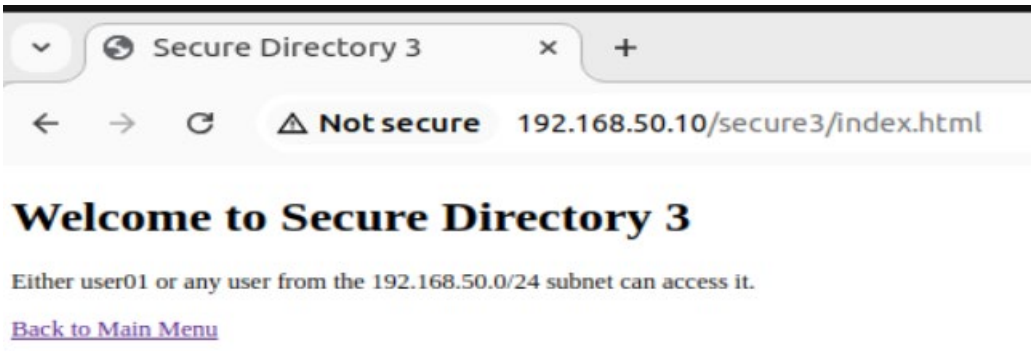
Test is done from ubuntu using chrome browser

4.6.3.1.1 secure3 (user01 or 192.168.50.0/24 - Accessible)

1. Open chrome
2. Set ip 192.168.50.10
3. See main page is displayed
4. Select first choice for secure 3:

   secure3 (user01 or 192.168.50.0/24 - Accessible)

5. Secure Directory 3 appears



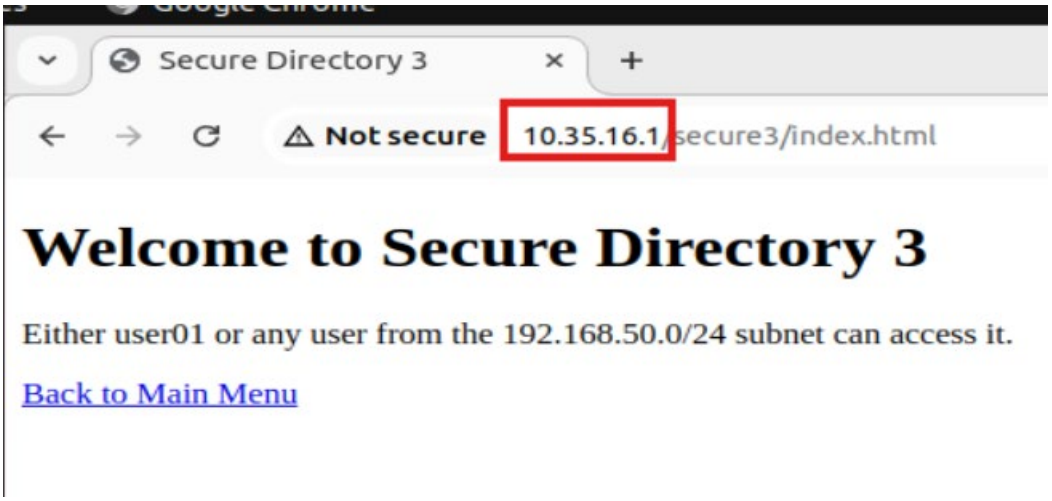
4.6.3.1.2 secure3 (user01 or 10.35.16.1/24 - Accessible)

1. Open chrome
2. Set ip 192.168.50.10
3. See main page is displayed
4. Select second choice for secure 3:

   secure3 (user01 or 10.35.16.1/24 - Accessible)

5. You will be prompted for user/password give user01/secret

6. Secure Directory 3 appears



4.7 Secure4

Create the directory and configuration to fulfill the following requirements:

Create a secure4 directory and configure it similarly to secure1 but grant access only to the user user02 (password “secret”).

4.7.1 Directories

1. Create directory

   mkdir /var/www/html_project1/secure4

   ```
   [root@server1 ~]# mkdir /var/www/html_project1/secure4
   ```

2. Verify directory is created

   ls -lqrtha /var/www/html_project1/

```
[root@server1 ~]# ls -lqrtha /var/www/html_project1/
total 32K
drwxr-xr-x. 5 root   root     54 Apr 16 14:31 ..
-rw-r--r--. 1 root   root    26K Apr 17 12:09 example.jpg
drwxr-xr-x. 2 apache apache   24 Apr 18 21:29 secure2
drwxr-xr-x. 2 apache apache   24 Apr 18 21:57 secure3
-rw-r--r--. 1 apache apache 2.0K Apr 18 22:41 index.html
drwxr-xr-x. 2 apache apache   24 Apr 18 23:57 secure1
drwxr-xr-x. 2 root   root      6 Apr 19 00:04 secure4
drwxr-xr-x. 6 apache apache  103 Apr 19 00:04 .
[root@server1 ~]#
```

4.7.2 Configuration

4.7.2.1 File httpd.conf

1. Modify configuration file */etc/httpd/conf/httpd.conf* to fulfill requirements
   Create a secure4 directory and configure it similarly to secure1 but grant access only to the user user02 (password “secret”).
2. Add the following lines to create a new Directory block

```
<Directory "/var/www/html_project1/secure4">
    AuthType Basic
    AuthName "Restricted Access - secure4"
    AuthUserFile /etc/httpd/.htpasswd
    Require user user02
</Directory>
```

a) Open the file for edition

vim /etc/httpd/conf/httpd.conf

```
<Directory "/var/www/html_project1/secure3">
    AuthType Basic
    AuthName "Restricted Access - secure3"
    AuthUserFile /etc/httpd/.htpasswd
    <RequireAny>
        Require user user01
        Require ip 192.168.50.0/24
    </RequireAny>
</Directory>

<Directory "/var/www/html_project1/secure4">
    AuthType Basic
    AuthName "Restricted Access - secure4"
    AuthUserFile /etc/httpd/.htpasswd
    Require user user02
</Directory>
```

b) Verify the syntax of configuration file after changes

httpd -t

```
[root@server1 conf]# httpd -t
Syntax OK
[root@server1 conf]#
```

c) Reload Apache:

systemctl reload httpd

```
[root@server1 ~]# systemctl reload httpd
[root@server1 ~]#
```

4.7.2.2 File index.html for secure4

1. Create index for secure4

   vim /var/www/html_project1/secure4/index.html

   Add the following lines

```
<!DOCTYPE html>
<html>
<head>
    <title>Secure Directory 4</title>
</head>
<body>
    <h1>Welcome to Secure Directory 4</h1>
    <p> grant access only to the user user02 (password “secret”)</p>
    <!-- Link to navigate back to the main menu -->
    <a href="http://192.168.50.10/index.html">Back to Main Menu</a>
</body>
</html>
```

```
<!DOCTYPE html>
<html>
<head>
    <title>Secure Directory 4</title>
</head>
<body>
    <h1>Welcome to Secure Directory 4</h1>
    <p> grant access only to the user user02 (password “secret”)</p>
    <!-- Link to navigate back to the main menu -->
    <a href="http://192.168.50.10/index.html">Back to Main Menu</a>
</body>
</html>
~
~
~
```

2. Change file permissions and ownership

   chown -R apache:apache /var/www/html_project1/secure4 ; chmod -R 755 /var/www/html_project1/secure4

```
[root@server1 ~]# chown -R apache:apache /var/www/html_project1/secure4 ; chmod -R 755 /var/www/html_project1/secure4
[root@server1 ~]#
```

3. Verify changes are applied

   ls -lqrtha /var/www/html_project1/secure4

```
[root@server1 ~]# ls -lqrtha /var/www/html_project1/secure4
total 4.0K
drwxr-xr-x. 6 apache apache 103 Apr 19 00:04 ..
-rwxr-xr-x. 1 apache apache 336 Apr 19 00:07 index.html
drwxr-xr-x. 2 apache apache  24 Apr 19 00:07 .
```

4.7.3 Test secure 4

4.7.3.1 Browser test

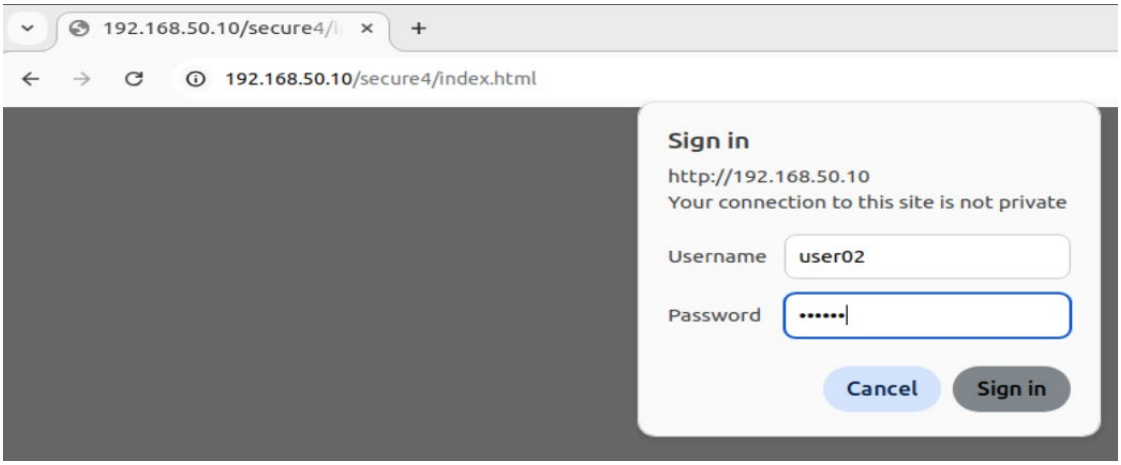
Test is done from ubuntu using chrome browser

4.7.3.1.1 secure4 (user02 - Accessible)

1. Open chrome
2. Set ip 192.168.50.10
3. See main page is displayed
4. Select second choice for secure 3:

   secure4 (user02 - Accessible)

5. You will be prompted for user/password give user02/secret



6. Secure Directory 4 appears

4.8 Secure5

Create the directories and configuration to fulfill the following requirements:

Create a secure5 directory and place an .htaccess file inside it to restrict access to only user01 from any subnet.

4.8.1 Directories

1. Create directory

   mkdir /var/www/html_project1/secure5

```
[root@server1 ~]# mkdir /var/www/html_project1/secure5
[root@server1 ~]#
```

   Verify directory is created

```
[root@server1 ~]# ls -lqrtha /var/www/html_project1/
total 32K
drwxr-xr-x. 5 root   root     54 Apr 16 14:31 ..
-rw-r--r--. 1 root   root    26K Apr 17 12:09 example.jpg
drwxr-xr-x. 2 apache apache   24 Apr 18 21:29 secure2
drwxr-xr-x. 2 apache apache   24 Apr 18 21:57 secure3
-rw-r--r--. 1 apache apache 2.0K Apr 18 22:41 index.html
drwxr-xr-x. 2 apache apache   24 Apr 18 23:57 secure1
drwxr-xr-x. 2 apache apache   24 Apr 19 00:07 secure4
drwxr-xr-x. 7 apache apache  118 Apr 19 00:32 .
drwxr-xr-x. 2 root   root      6 Apr 19 00:40 secure5
[root@server1 ~]#
```

4.8.2 Configuration

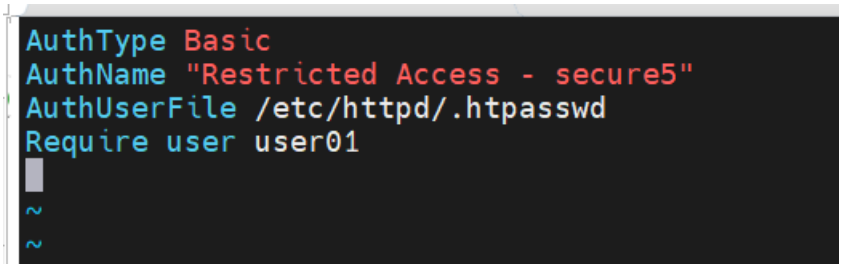
4.8.2.1 .httaccess

1. Create .htaccess file

   vim /var/www/html_project1/secure5/.htaccess

Add the following lines

```
AuthType Basic
AuthName "Restricted Access - secure5"
AuthUserFile /etc/httpd/.htpasswd
Require user user01
```

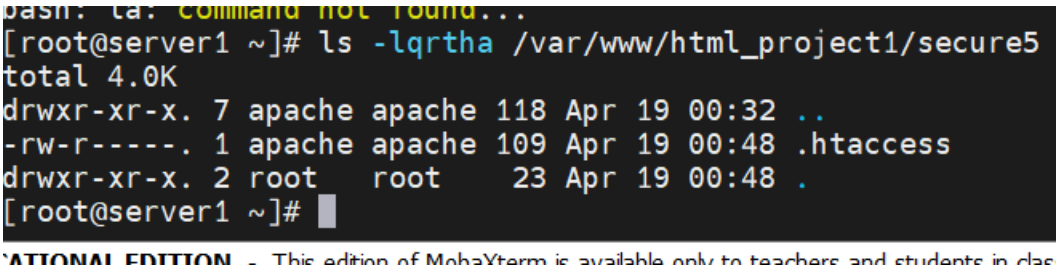


2. Change ownership and permissions

chown apache:apache /var/www/html_project1/secure5/.htaccess ; chmod 640 /var/www/html_project1/secure5/.htaccess

```
[root@server1 ~]# chown apache:apache /var/www/html_project1/secure5/.htaccess ; chmod 640 /var/www/html_project1/secure5/.htaccess
```

la -lqrtha /var/www/html_project1/secure5



4.8.2.2 File httpd.conf

1. Modify configuration file */etc/httpd/conf/httpd.conf* to fulfill requirements
   Create a secure5 directory and place an .htaccess file inside it to restrict access to only user01 from any subnet.
2. Add the following lines to create a new Directory block

   Enable .htaccess support

```
<Directory "/var/www/html_project1/secure5">
    AllowOverride All
</Directory>
```

d) Open the file for edition

vim /etc/httpd/conf/httpd.conf

```
</Directory>

<Directory "/var/www/html_project1/secure4">
    AuthType Basic
    AuthName "Restricted Access - secure4"
    AuthUserFile /etc/httpd/.htpasswd
    Require user user02
</Directory>

<Directory "/var/www/html_project1/secure5">
    AllowOverride All
</Directory>
```

e) Verify the syntax of configuration file after changes

httpd -t

```
[root@server1 conf]# httpd -t
Syntax OK
[root@server1 conf]#
```

f) Reload Apache:

systemctl reload httpd

```
[root@server1 ~]# systemctl reload httpd
[root@server1 ~]#
```

4.8.2.3 File index.html for secure5

1. Create index for secure5

   vim /var/www/html_project1/secure5/index.html

   Add the following lines

```
<!DOCTYPE html>
<html>
<head>
    <title>Secure Directory 5</title>
</head>
<body>
    <h1>Welcome to Secure Directory 5</h1>
    <p> Restrict access to only user01 from any subnet.</p>
    <!-- Link to navigate back to the main menu -->
    <a href="http://192.168.50.10/index.html">Back to Main Menu</a>
</body>
</html>
```

```
<!DOCTYPE html>
<html>
<head>
    <title>Secure Directory 5</title>
</head>
<body>
    <h1>Welcome to Secure Directory 5</h1>
    <p> Restrict access to only user01 from any subnet.</p>
    <!-- Link to navigate back to the main menu -->
    <a href="http://192.168.50.10/index.html">Back to Main Menu</a>
</body>
</html>
```

2. Change file permissions and ownership

   chown -R apache:apache /var/www/html_project1/secure5/index.html ; chmod -R 755 /var/www/html_project1/secure5/index.html

```
[root@server1 ~]# chown -R apache:apache /var/www/html_project1/secure5/index.html ; chmod -R 755 /var/www/html_project1/secure5/index.html
```

3. Verify changes are applied

   ls -lqrtha /var/www/html_project1/secure5

```
[root@server1 ~]# ls -lqrtha /var/www/html_project1/secure5
total 8.0K
drwxr-xr-x. 7 apache apache 118 Apr 19 00:32 ..
-rw-r-----. 1 apache apache 109 Apr 19 00:48 .htaccess
-rwxr-xr-x. 1 apache apache 323 Apr 19 01:05 index.html
drwxr-xr-x. 2 apache apache  41 Apr 19 01:05 .
[root@server1 ~]#
```

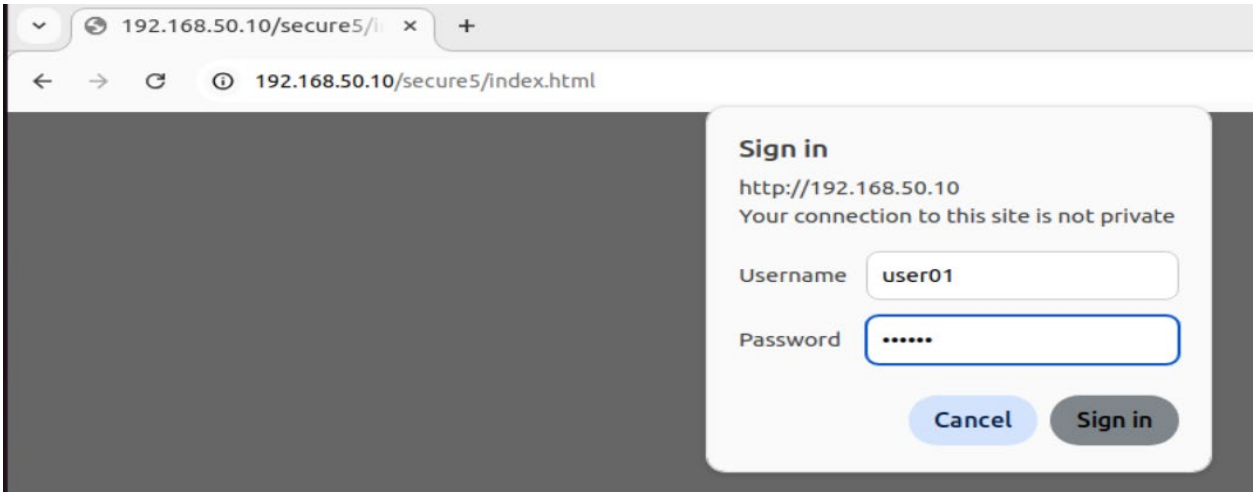
4.8.3 Test secure 5

4.8.3.1 Browser test

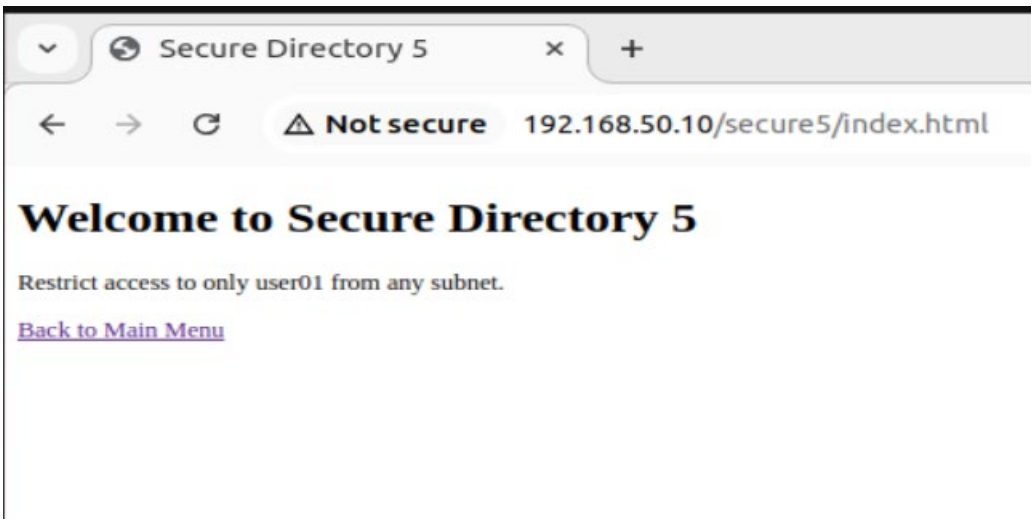
Test is done from ubuntu using chrome browser

4.8.3.1.1 secure5 (user01 with .htaccess - Accessible}

1. Open chrome
2. Set ip 192.168.50.10
3. See main page is displayed
4. Select choice for secure 5:
   secure5 (user01 with .htaccess - Accessible}
5. You will be prompted for user/password give user01/secret



6. Secure Directory 5 appears



4.9 Secure6

Create the directories and configuration to fulfill the following requirements:

Create a secure6 directory and use an .htaccess file to restrict access to user01, but only when connecting from the 192.168.50.0/24 subnet.

4.9.1 Directories

2. Create directory

   mkdir /var/www/html_project1/secure6

```
[root@server1 ~]# mkdir /var/www/html_project1/secure6
[root@server1 ~]#
```

   Verify directory is created

   ls -lqrtha /var/www/html_project1/

```
[root@server1 ~]# ls -lqrtha /var/www/html_project1/
total 32K
drwxr-xr-x. 5 root   root     54 Apr 16 14:31 ..
-rw-r--r--. 1 root   root    26K Apr 17 12:09 example.jpg
drwxr-xr-x. 2 apache apache   24 Apr 18 21:29 secure2
drwxr-xr-x. 2 apache apache   24 Apr 18 21:57 secure3
-rw-r--r--. 1 apache apache 2.0K Apr 18 22:41 index.html
drwxr-xr-x. 2 apache apache   24 Apr 18 23:57 secure1
drwxr-xr-x. 2 apache apache   24 Apr 19 00:07 secure4
drwxr-xr-x. 2 apache apache   41 Apr 19 01:05 secure5
drwxr-xr-x. 2 root   root      6 Apr 19 01:36 secure6
drwxr-xr-x. 8 apache apache  133 Apr 19 01:36 .
[root@server1 ~]#
```

4.9.2 Configuration

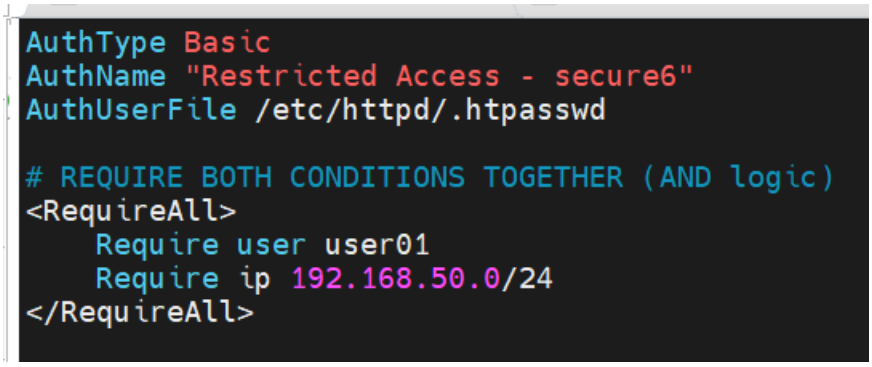
4.9.2.1 .httaccess

3. Create .htaccess file

vim /var/www/html_project1/secure6/.htaccess

Add the following lines

```
AuthType Basic
AuthName "Restricted Access - secure6"
AuthUserFile /etc/httpd/.htpasswd

# REQUIRE BOTH CONDITIONS TOGETHER (AND logic)
<RequireAll>
    Require user user01
    Require ip 192.168.50.0/24
</RequireAll>
```



4. Change ownership and permissions

chown apache:apache /var/www/html_project1/secure6/.htaccess ; chmod 640 /var/www/html_project1/secure6/.htaccess

```
[root@server1 ~]# vim /var/www/html_project1/secure6/.htaccess
[root@server1 ~]#
```

ls -lqrtha /var/www/html_project1/secure6

```
drwxr-xr-x. 7 apache apache 118 Apr 19 00:32 ..
-rw-r-----. 1 apache apache 109 Apr 19 00:48 .htaccess
drwxr-xr-x. 2 root   root    23 Apr 19 00:48 .
[root@server1 ~]#
```

4.9.2.2 File httpd.conf

1. Modify configuration file */etc/httpd/conf/httpd.conf* to fulfill requirements

   Create a secure6 directory and use an .htaccess file to restrict access to user01, but only when connecting from the 192.168.50.0/24 subnet.

2. Add the following lines to create a new Directory block

   Enable .htaccess support

```
<Directory "/var/www/html_project1/secure6">
    AllowOverride All
</Directory>
```

g) Open the file for edition

vim /etc/httpd/conf/httpd.conf

```
<Directory "/var/www/html_project1/secure5">
    AllowOverride All
</Directory>

<Directory "/var/www/html_project1/secure6">
    AllowOverride All
</Directory>

<Directory "/var/www/html_project1/secure7">
    AllowOverride All
```

h) Verify the syntax of configuration file after changes

httpd -t

```
[root@server1 conf]# httpd -t
Syntax OK
[root@server1 conf]#
```

i) Reload Apache:

systemctl reload httpd

```
[root@server1 ~]# systemctl reload httpd
[root@server1 ~]#
```

4.9.2.3 File index.html for secure6

4. Create index for secure6

vim /var/www/html_project1/secure6/index.html

```
[root@server1 ~]# vim  /var/www/html_project1/secure6/index.html
```

Add the following lines

```
<!DOCTYPE html>
<html>
<head>
    <title>Secure Directory 6</title>
</head>
<body>
    <h1>Welcome to Secure Directory 6</h1>
    <p> Restrict access to user01, allow only when connecting from the 192.168.50.0/24 subnet.</p>
    <!-- Link to navigate back to the main menu -->
    <a href="http://192.168.50.10/index.html">Back to Main Menu</a>
</body>
</html>
```

```
<!DOCTYPE html>
<html>
<head>
    <title>Secure Directory 6</title>
</head>
<body>
    <h1>Welcome to Secure Directory 6</h1>
    <p> Restrict access to user01, allow only when connecting from the 192.168.50.0/24 subnet.</p>
    <!-- Link to navigate back to the main menu -->
    <a href="http://192.168.50.10/index.html">Back to Main Menu</a>
</body>
</html>
~
~
```

5. Change file permissions and ownership

chown -R apache:apache /var/www/html_project1/secure6/index.html ; chmod -R 755 /var/www/html_project1/secure6/index.html

```
[root@server1 ~]# chown -R apache:apache /var/www/html_project1/secure6/index.html ; chmod -R 755 /var/www/html_project1/secure6/index.html
[root@server1 ~]#
```

6. Verify changes are applied

ls -lqrtha /var/www/html_project1/secure6

```
[root@server1 ~]# ls -lqrtha /var/www/html_project1/secure6
total 8.0K
drwxr-xr-x. 8 apache apache 133 Apr 19 01:36 ..
-rw-r-----. 1 apache apache 220 Apr 19 01:49 .htaccess
-rwxr-xr-x. 1 apache apache 363 Apr 19 01:58 index.html
drwxr-xr-x. 2 root   root    41 Apr 19 01:58 .
[root@server1 ~]#
```

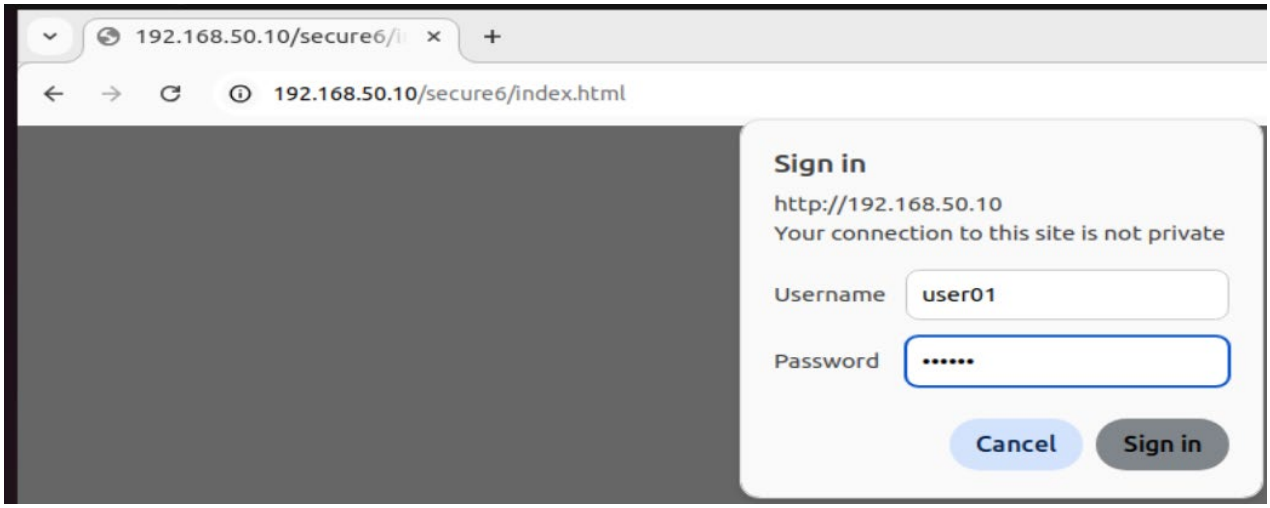
4.9.3 Test secure 6

4.9.3.1 Browser test

Test is done from ubuntu using chrome browser

4.9.3.1.1 secure6 (user01 and 192.168.50 with .htaccess - Accessible)

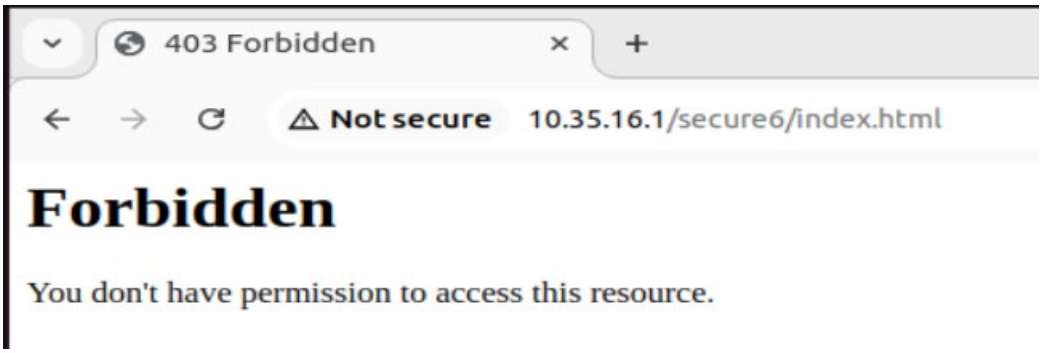
1. Open chrome
2. Set ip 192.168.50.10
3. See main page is displayed
4. Select first choice for secure 6:
   secure6 (user01 and 192.168.50 with .htaccess - Accessible)
5. You will be prompted for user/password give user01/secret



6. Secure Directory 6 appears

4.9.3.1.2 secure6 (user01 and 10.35.16.1/24 with .htaccess - Not accessible)

1. Open chrome
2. Set ip 192.168.50.10
3. See main page is displayed
4. Select first choice for secure 6:
   secure6 (user01 and 10.35.16.1/24 with .htaccess - Not accessible)
5. Error 403 Forbidden appears



4.10 Secure7

Create the directories and configuration to fulfill the following requirements:

Create a secure7 directory and use an .htaccess file to allow access to either the user01 or any user from the 192.168.50.0/24 subnet.

4.10.1 Directories

3. Create directory

   mkdir /var/www/html_project1/secure7

```
root@server1 ~]# mkdir /var/www/html_project1/secure7
root@server1 ~]#
```

Verify directory is created

ls -lqrtha /var/www/html_project1

```
[root@server1 ~]# ls -lqrtha /var/www/html_project1
total 32K
drwxr-xr-x. 5 root   root     54 Apr 16 14:31 ..
-rw-r--r--. 1 root   root    26K Apr 17 12:09 example.jpg
drwxr-xr-x. 2 apache apache   24 Apr 18 21:29 secure2
drwxr-xr-x. 2 apache apache   24 Apr 18 21:57 secure3
-rw-r--r--. 1 apache apache 2.0K Apr 18 22:41 index.html
drwxr-xr-x. 2 apache apache   24 Apr 18 23:57 secure1
drwxr-xr-x. 2 apache apache   24 Apr 19 00:07 secure4
drwxr-xr-x. 2 apache apache   41 Apr 19 01:05 secure5
drwxr-xr-x. 2 root   root     41 Apr 19 01:58 secure6
drwxr-xr-x. 2 root   root      6 Apr 19 02:24 secure7
drwxr-xr-x. 9 apache apache  148 Apr 19 02:24 .
[root@server1 ~]#
```

4.10.2 Configuration

4.10.2.1.httaccess

1. Create .htaccess file

vim /var/www/html_project1/secure7/.htaccess

```
[root@server1 ~]# vim /var/www/html_project1/secure7/.htaccess
[root@server1 ~]#
```

Add the following lines

```
AuthType Basic
AuthName "Restricted Access - secure7"
AuthUserFile /etc/httpd/.htpasswd

# ALLOW EITHER CONDITION (OR logic)
<RequireAny>
    Require user user01
    Require ip 192.168.50.0/24
</RequireAny>
```

```
AuthType Basic
AuthName "Restricted Access - secure7"
AuthUserFile /etc/httpd/.htpasswd

# ALLOW EITHER CONDITION (OR logic)
<RequireAny>
    Require user user01
    Require ip 192.168.50.0/24
</RequireAny>
~
~
~
```

2. Change ownership and permissions

chown apache:apache /var/www/html_project1/secure7/.htaccess ; chmod 640 /var/www/html_project1/secure7/.htaccess

```
[root@server1 ~]# chown apache:apache /var/www/html_project1/secure7/.htaccess  ;  chmod 640 /var/www/html_project1/secure7/.htaccess
```

la -lqrtha /var/www/html_project1/secure5

```
bash: la: command not found...
[root@server1 ~]# ls -lqrtha /var/www/html_project1/secure5
total 4.0K
drwxr-xr-x. 7 apache apache 118 Apr 19 00:32 ..
-rw-r-----. 1 apache apache 109 Apr 19 00:48 .htaccess
drwxr-xr-x. 2 root   root    23 Apr 19 00:48 .
[root@server1 ~]#
```

4.10.2.2File httpd.conf

1. Modify configuration file */etc/httpd/conf/httpd.conf* to fulfill requirements
   Create a secure5 directory and place an .htaccess file inside it to restrict access to only user01 from any subnet.
2. Add the following lines to create a new Directory block

   Enable .htaccess support

```
<Directory "/var/www/html_project1/secure7">
    AllowOverride All
</Directory>
```

a) Open the file for edition

vim /etc/httpd/conf/httpd.conf

```
<Directory "/var/www/html_project1/secure5">
    AllowOverride All
</Directory>


<Directory "/var/www/html_project1/secure6">
    AllowOverride All
</Directory>

<Directory "/var/www/html_project1/secure7">
    AllowOverride All
</Directory>

<IfModule dir_module>
    DirectoryIndex index.html
</IfModule>
```

b) Verify the syntax of configuration file after changes

httpd -t

```
[root@server1 conf]# httpd -t
Syntax OK
[root@server1 conf]#
```

c) Reload Apache:

systemctl reload httpd

```
[root@server1 ~]# systemctl reload httpd
[root@server1 ~]#
```

4.10.2.3File index.html for secure5

1. Create index for secure5

   vim /var/www/html_project1/secure7/index.html

   Add the following lines

```
<!DOCTYPE html>
<html>
<head>
    <title>Secure Directory 7</title>
</head>
<body>
    <h1>Welcome to Secure Directory 7</h1>
    <p>Allow access to either the
user01 or any user from the 192.168.50.0/24 subnet.</p>
    <!-- Link to navigate back to the main menu -->
    <a href="http://192.168.50.10/index.html">Back to Main Menu</a>
```

```
</body>
</html>
```

```
<!DOCTYPE html>
<html>
<head>
    <title>Secure Directory 7</title>
</head>
<body>
    <h1>Welcome to Secure Directory 7</h1>
    <p> Allow access to either the
user01 or any user from the 192.168.50.0/24 subnet.</p>
    <!-- Link to navigate back to the main menu -->
    <a href="http://192.168.50.10/index.html">Back to Main Menu</a>
</body>
</html>
```

2. Change file permissions and ownership

   chown -R apache:apache /var/www/html_project1/secure7/index.html ; chmod -R 755 /var/www/html_project1/secure7/index.html

```
[root@server1 ~]# chown -R apache:apache /var/www/html_project1/secure7/index.html ; chmod -R 755 /var/www/html_project1/secure7/index.html
[root@server1 ~]#
```

3. Verify changes are applied

   ls -lqrtha /var/www/html_project1/secure7

```
[root@server1 ~]# ls -lqrtha /var/www/html_project1/secure7
total 8.0K
drwxr-xr-x. 9 apache apache 148 Apr 19 02:24 ..
-rw-r-----. 1 apache apache 208 Apr 19 02:31 .htaccess
-rwxr-xr-x. 1 apache apache 356 Apr 19 02:37 index.html
drwxr-xr-x. 2 root   root    41 Apr 19 02:37 .
[root@server1 ~]#
```

4.10.3 Test secure 7

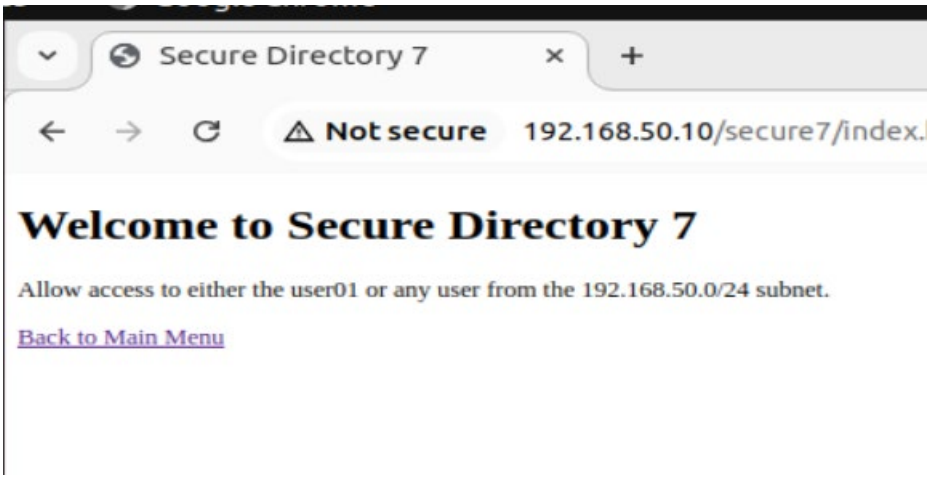
4.10.3.1 Browser test

Test is done from ubuntu using chrome browser

4.10.3.1.1 secure7 (user01 or 192.168.50 with .htaccess -Accessible)

1. Open chrome
2. Set ip 192.168.50.10
3. See main page is displayed
4. Select choice for secure 7:
   secure7 (user01 or 192.168.50 with .htaccess -Accessible)

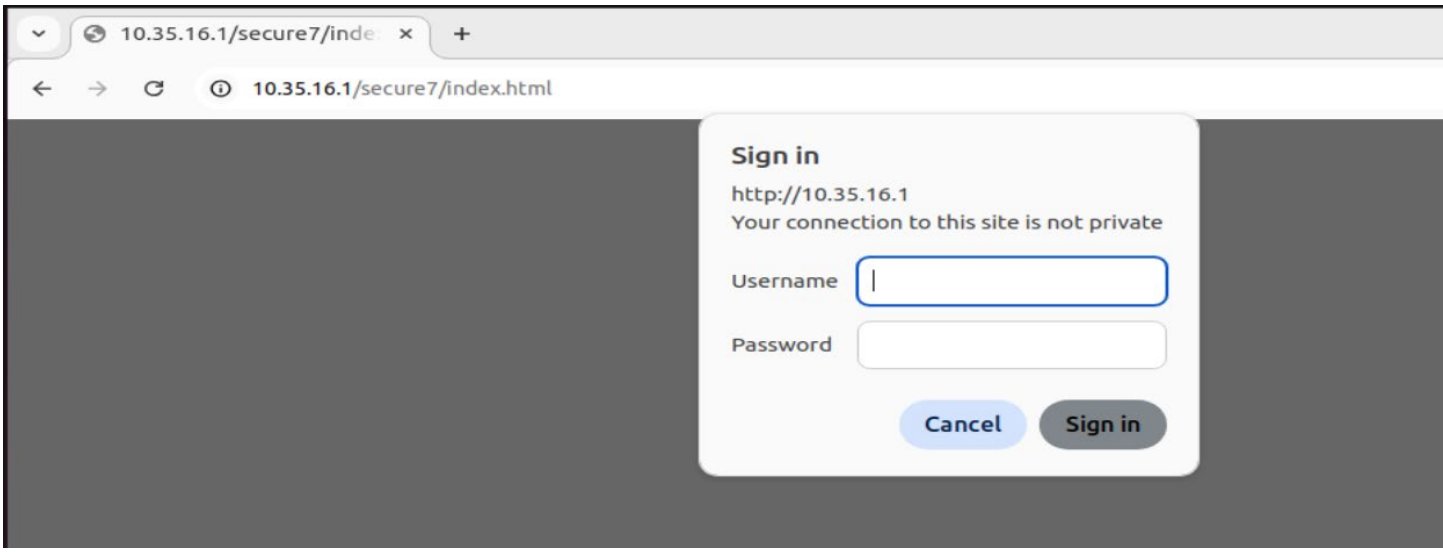
5. Secure Directory 7 appears



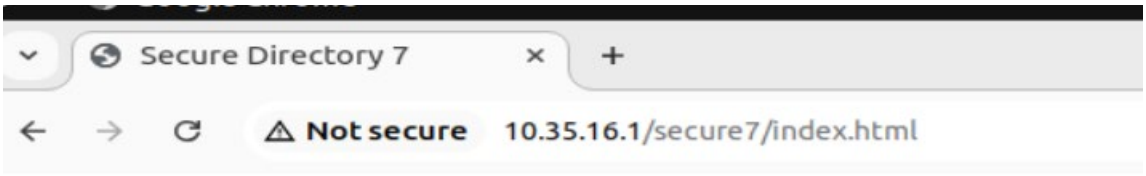
4.10.3.1.2 secure7 (user01 or 10.35.16.1 with .htaccess - Accessible)

1. Open chrome
2. Set ip 192.168.50.10
3. See main page is displayed
4. Select choice for secure 5:

   secure7 (user01 or 10.35.16.1 with .htaccess - Accessible)
5. You will be prompted for user/password give user01/secret



6. Secure Directory 7 appears



Welcome to Secure Directory 7

Allow access to either the user01 or any user from the 192.168.50.0/24 subnet.

Back to Main Menu

5 TASK 3 – ACCESSIBILITY

5.1 Requirements

1. In the**/var/www/html_project1** directory add the following subdirectories:
   - **Project1**
   - **Project2**
   - **Project3**
   - **Project4**
2. In each of these directories, create a web page named after the directory itself. For example, **Project1/project1.html**, **Project2/project2.html,** etc.
3. **Do not place an index.html file** in these directories. Instead, configure Apache to display a **directory listing** when accessed.
4. Add <Directory> and <Files> directives to implement the following access controls:
   - All directories and their contents must be accessible from the **192.168.50.0/24** subnet.
   - **Project1**: Accessible only from the **10.35.16.0/24** and **10.35.17.0/24** subnets. Any files named **secret.*** must not be accessible.
   - **Project2**: Not accessible only from the **10.35.16.0/24** subnet. All files must be accessible to others.
   - **Project3:** Accessible only from the **192.168.100.0/24** subnet. All ***.gif** files must not be accessible.
   - **Project4:** Accessible only from **10.35.16.0/24** and **192.168.100.0/24.** All **test.html** files must not be accessible.
5. For all directories, ***.txt** files must not be accessible. Place the corresponding **directive outside** of any **<Directory>** blocks.

5.2 Modify main html

Modify main html file to include Task 3 menu

Open the file for edition and include the lines in the box, related to Task 3

vi /var/www/html_project1/index.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Project Part I - Homepage</title>
</head>
<body>
    <p><b><i><u>Welcome to Project Part I</u></i></b></p>
    <hr>

    <!-- Task 2 - Secure Directories -->
    <h2>Task 2 - Secure Directories</h2>
    <ul>
        <li><a href="http://192.168.50.10/secure1/index.html">Secure1 (user01 - Accessible)</a></li>
        <li><a href="http://192.168.50.10/secure2/index.html">Secure2 (user01 and 192.168.50.0/24 -
Accessible)</a></li>
        <li><a href="http://10.35.16.1/secure2/index.html">Secure2 (user01 and 10.35.16.1/24 - Not
Accessible)</a></li>
```

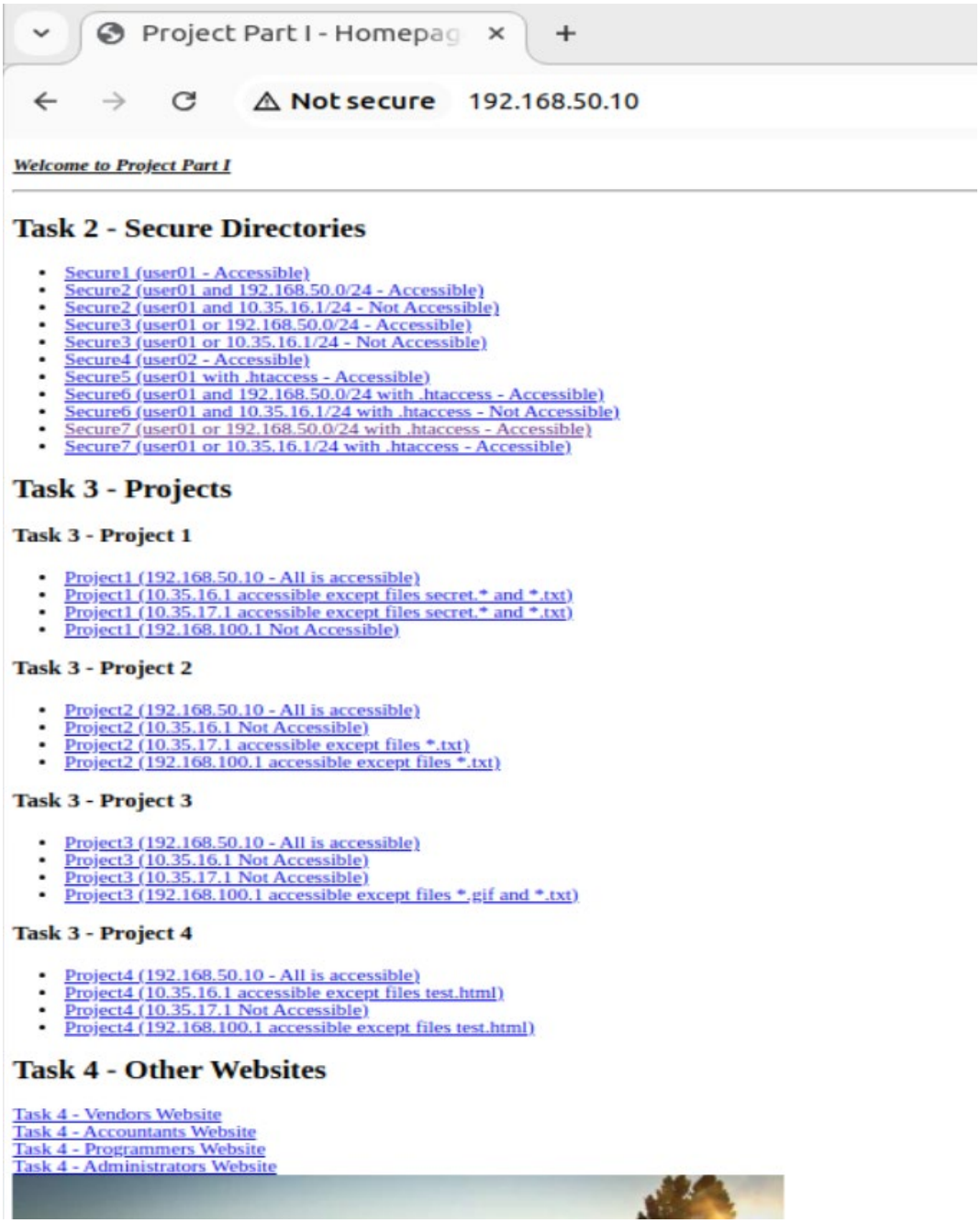
```
        <li><a href="http://192.168.50.10/secure3/index.html">Secure3 (user01 or 192.168.50.0/24 -
Accessible)</a></li>
        <li><a href="http://10.35.16.1/secure3/index.html">Secure3 (user01 or 10.35.16.1/24 - Not
Accessible)</a></li>
        <li><a href="http://192.168.50.10/secure4/index.html">Secure4 (user02 - Accessible)</a></li>
        <li><a href="http://192.168.50.10/secure5/index.html">Secure5 (user01 with .htaccess -
Accessible)</a></li>
        <li><a href="http://192.168.50.10/secure6/index.html">Secure6 (user01 and 192.168.50.0/24 with
.htaccess - Accessible)</a></li>
        <li><a href="http://10.35.16.1/secure6/index.html">Secure6 (user01 and 10.35.16.1/24 with .htaccess -
Not Accessible)</a></li>
        <li><a href="http://192.168.50.10/secure7/index.html">Secure7 (user01 or 192.168.50.0/24 with
.htaccess - Accessible)</a></li>
        <li><a href="http://10.35.16.1/secure7/index.html">Secure7 (user01 or 10.35.16.1/24 with .htaccess -
Accessible)</a></li>
    </ul>

    <!-- Task 3 - Projects -->
    <h2>Task 3 - Projects</h2>

    <!-- Task 3 - Project 1 -->
    <h3>Task 3 - Project 1</h3>
    <ul>
        <li><a href="http://192.168.50.10/Project1/">Project1 (192.168.50.10 - All is accessible)</a></li>
        <li><a href="http://10.35.16.1/Project1/">Project1 (10.35.16.1 accessible except files secret.* and
*.txt)</a></li>
        <li><a href="http://10.35.17.1/Project1/">Project1 (10.35.17.1 accessible except files secret.* and
*.txt)</a></li>
        <li><a href="http://192.168.100.1/Project1/">Project1 (192.168.100.1 Not Accessible)</a></li>
    </ul>

    <!-- Task 3 - Project 2 -->
    <h3>Task 3 - Project 2</h3>
    <ul>
        <li><a href="http://192.168.50.10/Project2/">Project2 (192.168.50.10 - All is accessible)</a></li>
        <li><a href="http://10.35.16.1/Project2/">Project2 (10.35.16.1 Not Accessible)</a></li>
        <li><a href="http://10.35.17.1/Project2/">Project2 (10.35.17.1 accessible except files
*.txt)</a></li>
        <li><a href="http://192.168.100.1/Project2/">Project2 (192.168.100.1 accessible except files
*.txt)</a></li>
    </ul>

    <!-- Task 3 - Project 3 -->
    <h3>Task 3 - Project 3</h3>
    <ul>
        <li><a href="http://192.168.50.10/Project3/">Project3 (192.168.50.10 - All is accessible)</a></li>
        <li><a href="http://10.35.16.1/Project3/">Project3 (10.35.16.1 Not Accessible)</a></li>
        <li><a href="http://10.35.17.1/Project3/">Project3 (10.35.17.1 Not Accessible)</a></li>
        <li><a href="http://192.168.100.1/Project3/">Project3 (192.168.100.1 accessible except files *.gif
and *.txt)</a></li>
    </ul>

    <!-- Task 3 - Project 4 -->
    <h3>Task 3 - Project 4</h3>
    <ul>
        <li><a href="http://192.168.50.10/Project4/">Project4 (192.168.50.10 - All is accessible)</a></li>
        <li><a href="http://10.35.16.1/Project4/">Project4 (10.35.16.1 accessible except files
test.html)</a></li>
        <li><a href="http://10.35.17.1/Project4/">Project4 (10.35.17.1 Not Accessible)</a></li>
        <li><a href="http://192.168.100.1/Project4/">Project4 (192.168.100.1 accessible except files
test.html)</a></li>
    </ul>

    <!-- Task 4 - Other Websites -->
    <h2>Task 4 - Other Websites</h2>
    <a href="vendors.html">Task 4 - Vendors Website</a><br>
    <a href="accountants.html">Task 4 - Accountants Website</a><br>
    <a href="programmers.html">Task 4 - Programmers Website</a><br>
    <a href="administrators.html">Task 4 - Administrators Website</a><br>
    <img src="example.jpg" alt="Example Image"><br>
</body>
</html>
```

Note as per requirement :

Do not place an index.html file in these directories. Instead, configure Apache to display a directory listing when accessed.

The format href="http://<ip_address>/Projectx/" is used.

Verify web page display (links still not working)

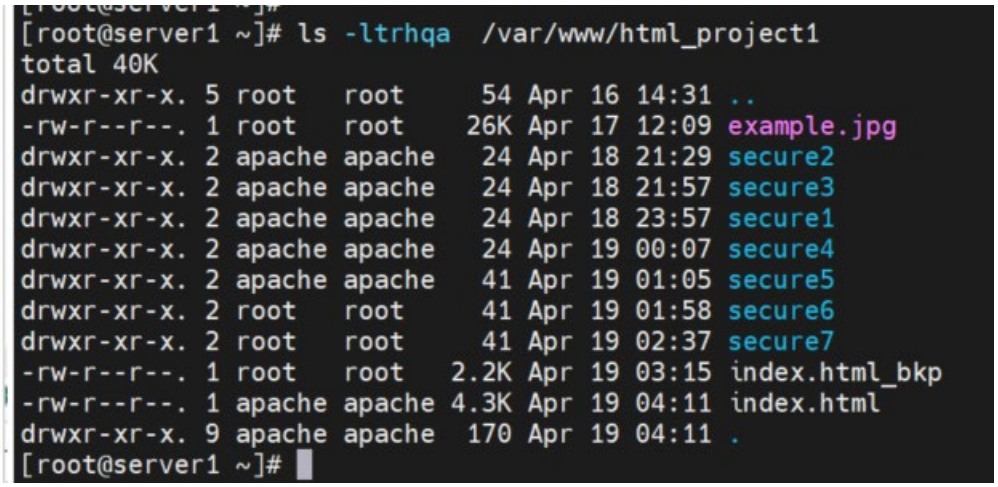


5.3 Create Directory Structure and Web Pages

5.3.1 Create project subdirectories

1. List main directory

   ls -ltrhqa /var/www/html_project1



2. Create project subdirectories

   mkdir /var/www/html_project1/Project1
   mkdir /var/www/html_project1/Project2
   mkdir /var/www/html_project1/Project3
   mkdir /var/www/html_project1/Project4

3. List main directory

   ls -ltrhqa /var/www/html_project1

```
[root@server1 ~]# ls -ltrhqa  /var/www/html_project1
total 44K
drwxr-xr-x.  5 root   root     54 Apr 16 14:31 ..
-rw-r--r--.  1 root   root    26K Apr 17 12:09 example.jpg
drwxr-xr-x.  2 apache apache   24 Apr 18 21:29 secure2
drwxr-xr-x.  2 apache apache   24 Apr 18 21:57 secure3
drwxr-xr-x.  2 apache apache   24 Apr 18 23:57 secure1
drwxr-xr-x.  2 apache apache   24 Apr 19 00:07 secure4
drwxr-xr-x.  2 apache apache   41 Apr 19 01:05 secure5
drwxr-xr-x.  2 root   root     41 Apr 19 01:58 secure6
drwxr-xr-x.  2 root   root     41 Apr 19 02:37 secure7
-rw-r--r--.  1 root   root   2.2K Apr 19 03:15 index.html_bkp
-rw-r--r--.  1 apache apache 4.3K Apr 19 04:11 index.html
drwxr-xr-x.  2 root   root      6 Apr 19 23:59 Project1
drwxr-xr-x.  2 root   root      6 Apr 19 23:59 Project2
drwxr-xr-x.  2 root   root      6 Apr 19 23:59 Project3
drwxr-xr-x.  2 root   root      6 Apr 19 23:59 Project4
drwxr-xr-x. 13 apache apache 4.0K Apr 19 23:59 .
[root@server1 ~]#
```

5.3.2 Create web pages in each directory

1. Project 1

   sudo sh -c 'echo "<html><body><h1>Project1 Page</h1></body></html>" > /var/www/html_project1/Project1/project1.html'

   cat /var/www/html_project1/Project1/project1.html

```
[root@server1 ~]#
[root@server1 ~]# sudo sh -c 'echo "<html><body><h1>Project1 Page</h1></body></html>" > /var/www/html_project1/Project1/project1.html'
[root@server1 ~]# cat /var/www/html_project1/Project1/project1.html
<html><body><h1>Project1 Page</h1></body></html>
[root@server1 ~]#
```

2. Project 2

   sudo sh -c 'echo "<html><body><h1>Project2 Page</h1></body></html>" > /var/www/html_project1/Project2/project2.html'

   cat /var/www/html_project1/Project2/project2.html

```
[root@server1 ~]#
[root@server1 ~]# sudo sh -c 'echo "<html><body><h1>Project2 Page</h1></body></html>" > /var/www/html_project1/Project2/project2.html'
[root@server1 ~]#
[root@server1 ~]# cat /var/www/html_project1/Project2/project2.html
<html><body><h1>Project2 Page</h1></body></html>
[root@server1 ~]#
```

3. Project 3

   sudo sh -c 'echo "<html><body><h1>Project3 Page</h1></body></html>" > /var/www/html_project1/Project3/project3.html'

   cat /var/www/html_project1/Project3/project3.html

```
[root@server1 ~]#
[root@server1 ~]# sudo sh -c 'echo "<html><body><h1>Project3 Page</h1></body></html>" > /var/www/html_project1/Project3/project3.html'
[root@server1 ~]# cat /var/www/html_project1/Project3/project3.html
<html><body><h1>Project3 Page</h1></body></html>
[root@server1 ~]#
```

4. Project 4

   sudo sh -c 'echo "<html><body><h1>Project4 Page</h1></body></html>" > /var/www/html_project1/Project4/project4.html'

   cat /var/www/html_project1/Project4/project4.html

```
[root@server1 ~]# sudo sh -c 'echo "<html><body><h1>Project4 Page</h1></body></html>" > /var/www/html_project1/Project4/project4.html'
[root@server1 ~]#
[root@server1 ~]# cat /var/www/html_project1/Project4/project4.html
<html><body><h1>Project4 Page</h1></body></html>
[root@server1 ~]#
```

5.3.3 Updated Configuration for /etc/httpd/conf/httpd.conf

1. Write directives according to the requiremnts

   Add these directives just before the </IfModule> closing tag for dir_module

```
# Project directories configuration

# Project1 Configuration
<Directory "/var/www/html_project1/Project1">
    Options +Indexes
    IndexOptions FancyIndexing
    AllowOverride None
    <RequireAny>
        Require ip 10.35.16.0/24
        Require ip 10.35.17.0/24
        Require ip 192.168.50.0/24
    </RequireAny>
    <Files "secret.*">
        Require all denied
    </Files>
</Directory>

# Project2 Configuration
<Directory "/var/www/html_project1/Project2">
    Options +Indexes
    IndexOptions FancyIndexing
    AllowOverride None
    <RequireAll>
        Require all granted
        Require not ip 10.35.16.0/24
    </RequireAll>
</Directory>

# Project3 Configuration
<Directory "/var/www/html_project1/Project3">
    Options +Indexes
    IndexOptions FancyIndexing
    AllowOverride None
    <RequireAny>
        Require ip 192.168.100.0/24
        Require ip 192.168.50.0/24
```

```
    </RequireAny>
    <Files "*.gif">
        Require all denied
    </Files>
</Directory>

# Project4 Configuration
<Directory "/var/www/html_project1/Project4">
    Options +Indexes
    IndexOptions FancyIndexing
    AllowOverride None
    <RequireAny>
        Require ip 10.35.16.0/24
        Require ip 192.168.100.0/24
        Require ip 192.168.50.0/24
    </RequireAny>
    <Files "test.html">
        Require all denied
    </Files>
</Directory>

# Global restriction for all .txt files across all directories
<Files "*.txt">
    Require all denied
</Files>
```

vim /etc/httpd/conf/httpd.conf

```
# Project directories configuration

# Project1 Configuration
<Directory "/var/www/html_project1/Project1">
    Options +Indexes
    IndexOptions FancyIndexing
    AllowOverride None
    <RequireAny>
        Require ip 10.35.16.0/24
        Require ip 10.35.17.0/24
        Require ip 192.168.50.0/24
    </RequireAny>
    <Files "secret.*">
        Require all denied
    </Files>
</Directory>

# Project2 Configuration
<Directory "/var/www/html_project1/Project2">
    Options +Indexes
    IndexOptions FancyIndexing
    AllowOverride None
    <RequireAll>
        Require all granted
        Require not ip 10.35.16.0/24
    </RequireAll>
</Directory>
```

```
</Directory>

# Project3 Configuration
<Directory "/var/www/html_project1/Project3">
    Options +Indexes
    IndexOptions FancyIndexing
    AllowOverride None
    <RequireAny>
        Require ip 192.168.100.0/24
        Require ip 192.168.50.0/24
    </RequireAny>
    <Files "*.gif">
        Require all denied
    </Files>
</Directory>

# Project4 Configuration
<Directory "/var/www/html_project1/Project4">
    Options +Indexes
    IndexOptions FancyIndexing
    AllowOverride None
    <RequireAny>
        Require ip 10.35.16.0/24
        Require ip 192.168.100.0/24
        Require ip 192.168.50.0/24
    </RequireAny>
    <Files "test.html">
        Require all denied
    </Files>
</Directory>

# Global restriction for all .txt files across all directories
<Files "*.txt">
    Require all denied
</Files>

<IfModule dir_module>
```

2. Verify config

   httpd -t

```
[root@server1 ~]# httpd -t
Syntax OK
[root@server1 ~]#
```

3. Restart httpd

   sudo systemctl restart httpd

   sudo systemctl status httpd

```
[root@server1 Documents]# sudo systemctl restart httpd
[root@server1 Documents]# sudo systemctl status httpd
● httpd.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
     Active: active (running) since Sun 2025-04-20 17:15:40 EDT; 6s ago
       Docs: man:httpd.service(8)
   Main PID: 133828 (httpd)
     Status: "Started, listening on: port 80"
      Tasks: 177 (limit: 22829)
     Memory: 45.4M
        CPU: 340ms
     CGroup: /system.slice/httpd.service
             ├─133828 /usr/sbin/httpd -DFOREGROUND
             ├─133830 /usr/sbin/httpd -DFOREGROUND
             ├─133831 /usr/sbin/httpd -DFOREGROUND
             ├─133832 /usr/sbin/httpd -DFOREGROUND
             └─133833 /usr/sbin/httpd -DFOREGROUND

Apr 20 17:15:40 server1 systemd[1]: Starting The Apache HTTP Server...
Apr 20 17:15:40 server1 systemd[1]: Started The Apache HTTP Server.
Apr 20 17:15:40 server1 httpd[133828]: Server configured, listening on: port 80
[root@server1 Documents]#
```

5.3.4 Create test files

1. Create test files using a bash script
   A) Create Project1 test files

/var/www/html_project1/Project1/P1_secret.txt
/var/www/html_project1/Project1/P1_secret.html
/var/www/html_project1/Project1/P1_secret.doc
/var/www/html_project1/Project1/P1_notes.txt
/var/www/html_project1/Project1/P1_image.gif
/var/www/html_project1/Project1/P1_logo.gif
/var/www/html_project1/Project1/secret.txt
/var/www/html_project1/Project1/secret.doc
/var/www/html_project1/Project1/test.html

   B) Create Project2 test files

/var/www/html_project1/Project2/P2_secret.txt
/var/www/html_project1/Project2/P2_secret.html
/var/www/html_project1/Project2/P2_secret.doc
/var/www/html_project1/Project2/P2_notes.txt
/var/www/html_project1/Project2/P2_image.gif
/var/www/html_project1/Project2/P2_logo.gif
/var/www/html_project1/Project2/secret.txt
/var/www/html_project1/Project2/secret.doc
/var/www/html_project1/Project2/test.html

C) Create Project3 test files

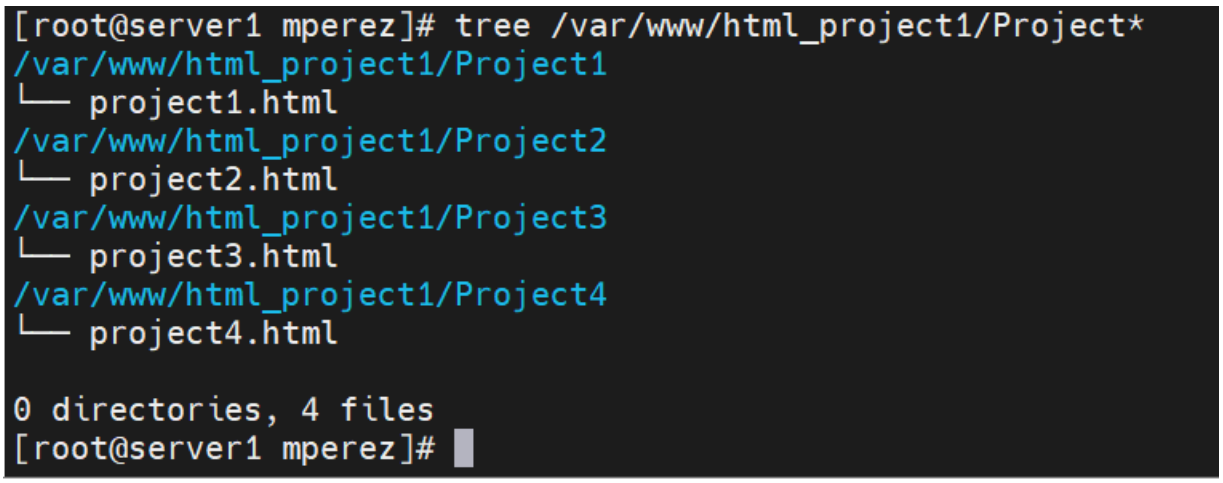
/var/www/html_project1/Project3/P3_secret.txt
/var/www/html_project1/Project3/P3_secret.html
/var/www/html_project1/Project3/P3_secret.doc
/var/www/html_project1/Project3/P3_notes.txt
/var/www/html_project1/Project3/P3_image.gif
/var/www/html_project1/Project3/P3_logo.gif
/var/www/html_project1/Project3/secret.txt
/var/www/html_project1/Project3/secret.doc
/var/www/html_project1/Project3/test.html

D) Create Project4 test files

/var/www/html_project1/Project4/P4_secret.txt
/var/www/html_project1/Project4/P4_secret.html
/var/www/html_project1/Project4/P4_secret.doc
/var/www/html_project1/Project4/P4_notes.txt
/var/www/html_project1/Project4/P4_image.gif
/var/www/html_project1/Project4/P4_logo.gif
/var/www/html_project1/Project4/secret.txt
/var/www/html_project1/Project4/secret.doc
/var/www/html_project1/Project4/test.html

2. Verify contents of current directory

tree /var/www/html_project1/Project*



3. Create script to create text files

vim /home/mperez/Documents/create_files.sh

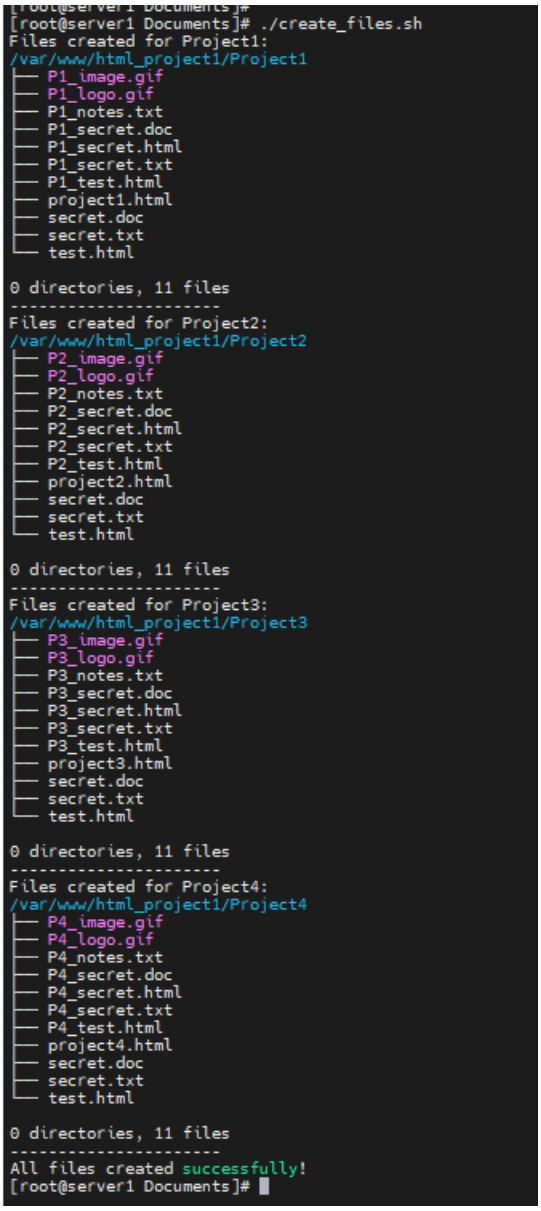
```
#!/bin/bash

# Define base directory
BASE_DIR="/var/www/html_project1"

# Array to hold project names
PROJECTS=("Project1" "Project2" "Project3" "Project4")

# Files to create for each project
FILES=(
    "secret.txt"
    "secret.html"
    "secret.doc"
    "notes.txt"
    "image.gif"
    "logo.gif"
    "test.html"
)

# Function to create files for a project
create_project_files() {
    PROJECT_NAME=$1
    PREFIX=$2

    # Create project directory if it doesn't exist
    mkdir -p "${BASE_DIR}/${PROJECT_NAME}"

    # Loop through each file and create it
    for FILE in "${FILES[@]}"; do
        touch
"${BASE_DIR}/${PROJECT_NAME}/${PREFIX}_${FILE}"
    done

    # Ensure the non-prefixed `secret.*` and
`test.html` files are created
    touch "${BASE_DIR}/${PROJECT_NAME}/secret.txt"
    touch "${BASE_DIR}/${PROJECT_NAME}/secret.doc"
    touch "${BASE_DIR}/${PROJECT_NAME}/test.html"

    # Display the directory structure
    echo "Files created for ${PROJECT_NAME}:"
    tree "${BASE_DIR}/${PROJECT_NAME}"
    echo "----------------------"
}

# Create files for each project with prefixes
create_project_files "Project1" "P1"
create_project_files "Project2" "P2"
create_project_files "Project3" "P3"
create_project_files "Project4" "P4"

echo "All files created successfully!"
```

4. Change the permissions of file to be excutable

   chmod 777 /home/mperez/Documents/create_files.sh

5. Execute created script

   cd /home/mperez/Documents/
   pwd
   ./create_files.sh



6. Verify created directories

   tree /var/www/html_project1/Project*

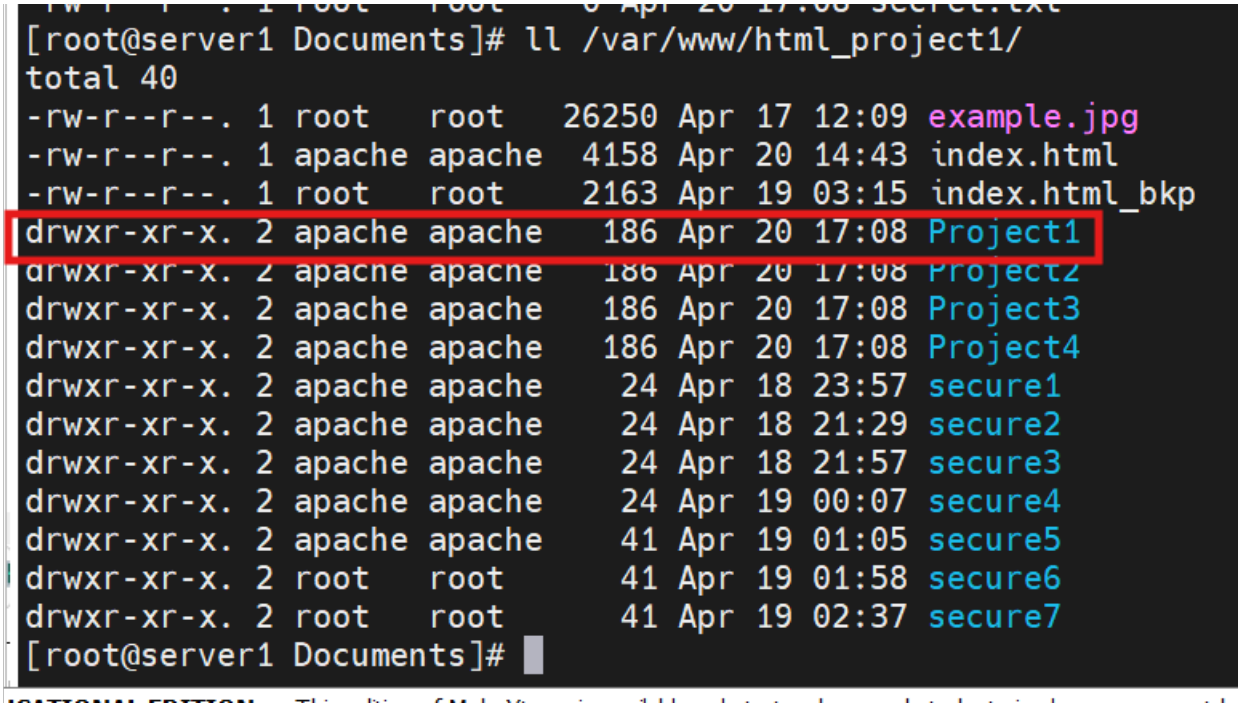
```
All files created successfully!
[root@server1 Documents]# tree /var/www/html_project1/Project*
/var/www/html_project1/Project1
├── P1_image.gif
├── P1_logo.gif
├── P1_notes.txt
├── P1_secret.doc
├── P1_secret.html
├── P1_secret.txt
├── P1_test.html
├── project1.html
├── secret.doc
├── secret.txt
└── test.html
/var/www/html_project1/Project2
├── P2_image.gif
├── P2_logo.gif
├── P2_notes.txt
├── P2_secret.doc
├── P2_secret.html
├── P2_secret.txt
├── P2_test.html
├── project2.html
├── secret.doc
├── secret.txt
└── test.html
/var/www/html_project1/Project3
├── P3_image.gif
├── P3_logo.gif
├── P3_notes.txt
├── P3_secret.doc
├── P3_secret.html
├── P3_secret.txt
├── P3_test.html
├── project3.html
├── secret.doc
├── secret.txt
└── test.html
/var/www/html_project1/Project4
├── P4_image.gif
├── P4_logo.gif
├── P4_notes.txt
├── P4_secret.doc
├── P4_secret.html
├── P4_secret.txt
├── P4_test.html
├── project4.html
├── secret.doc
├── secret.txt
└── test.html

0 directories, 44 files
[root@server1 Documents]#
```

5.3.5 Testing

5.3.5.1 Project1 testing

Requirement related to Project 1 are tested

1. In the/var/www/html_project1 directory add the following subdirectories: Project1

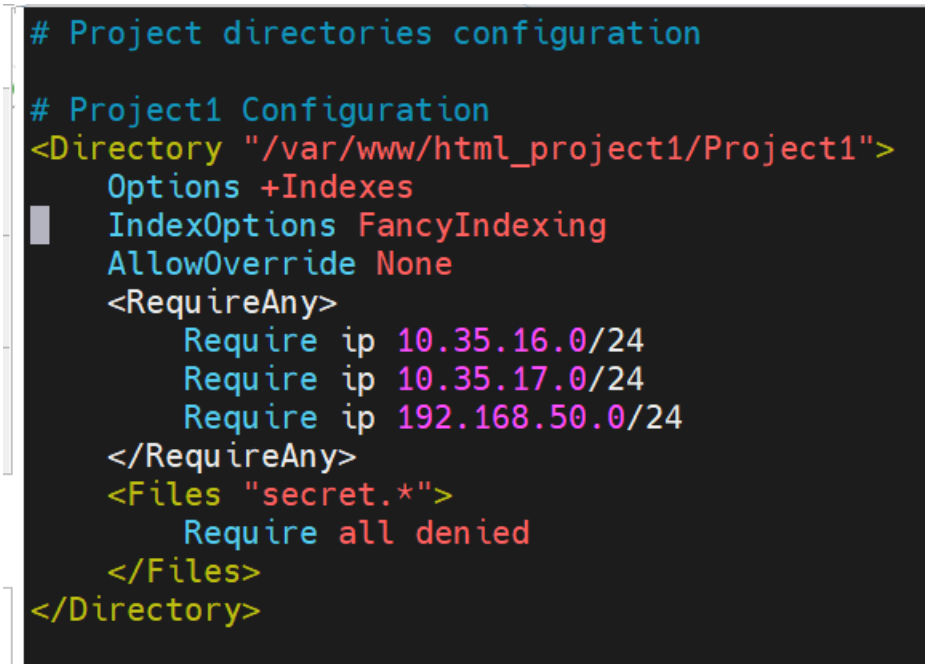


2. In each of these directories, create a web page named after the directory itself. For example, Project1/project1.html

ll /var/www/html_project1/Project1/p*.html

```
[root@server1 Documents]# ll /var/www/html_project1/Project1/p*.html
-rw-r--r--. 1 apache apache 49 Apr 20 00:55 /var/www/html_project1/Project1/project1.html
[root@server1 Documents]#
```

3. Requirements related to httpd configuration



```
# Project directories configuration

# Project1 Configuration
<Directory "/var/www/html_project1/Project1">
    Options +Indexes
    IndexOptions FancyIndexing
    AllowOverride None
    <RequireAny>
        Require ip 10.35.16.0/24
        Require ip 10.35.17.0/24
        Require ip 192.168.50.0/24
    </RequireAny>
    <Files "secret.*">
        Require all denied
    </Files>
</Directory>
```

Directive Explanation:

<Directory "/var/www/html_project1/Project1">**

This directive applies rules to the directory `/var/www/html_project1/Project1`. All configurations within this block affect the files and subdirectories under this path.

Options +Indexes

The `Options +Indexes` directive enables directory listing.
If a user accesses this directory without specifying a file (e.g., `http://server/Project1/`), Apache generates and displays an index of the files in the directory.

IndexOptions FancyIndexing

This configures the appearance of the directory index generated by `+Indexes`.
`FancyIndexing` adds visual enhancements to the directory listing, such as:

- Icons for file types.
- File sorting options.

AllowOverride None

This directive disables the use of `.htaccess` files in the directory. It ensures that settings in `.htaccess` are ignored, and only configurations explicitly set in the main Apache configuration file are applied. It is a security measure to prevent unauthorized overrides of server directives.

<RequireAny>

This block combines multiple conditions using "OR" logic. Access to this directory is granted if any one of the following `Require ip` directives is satisfied:

- `Require ip 10.35.16.0/24`: Clients from the `10.35.16.0/24` subnet are allowed.
- `Require ip 10.35.17.0/24`: Clients from the `10.35.17.0/24` subnet are allowed.
- `Require ip 192.168.50.0/24`: Clients from the `192.168.50.0/24` subnet are allowed.

This setup restricts access to only the specified subnets.

<Files "secret.*">

This block applies rules specifically to files matching the pattern `secret.*` (e.g., `secret.txt`, `secret.html`):

- `Require all denied`: Completely denies access to these files, regardless of the client's IP or any other rule.

It is a security measure to protect sensitive files from being viewed or downloaded.

Httpd configuration behavior summary

1. Users from the subnets `10.35.16.0/24`, `10.35.17.0/24`, or `192.168.50.0/24` can access the `/Project1` directory.
   Complies with requirements
   - All directories and their contents must be accessible from the 192.168.50.0/24 subnet.
   - Project1 is accessible only from the 10.35.16.0/24 and 10.35.17.0/24 subnets

2. Directory listing is enabled and styled with `FancyIndexing`.
   - Do not place an index.html file in these directories. Instead, configure Apache to display a directory listing when accessed.

3. `.htaccess` files are ignored to enforce central control over the directory's behavior.

   No .httaccess are used

4. Files matching the name pattern `secret.*` are blocked entirely, ensuring their confidentiality.
   - Any files named secret.* must not be accessible.

5. For all directories, *.txt files must not be accessible

```
# Global restriction for all .txt files across all directories
<Files "*.txt">
    Require all denied
</Files>
```

Project 1 files

tree /var/www/html_project1/Project1/

```
[root@server1 Documents]# tree /var/www/html_project1/Project1/
/var/www/html_project1/Project1/
├── P1_image.gif
├── P1_logo.gif
├── P1_notes.txt
├── P1_secret.doc
├── P1_secret.html
├── P1_secret.txt
├── P1_test.html
├── project1.html
├── secret.doc
├── secret.txt
└── test.html

0 directories, 11 files
[root@server1 Documents]#
```

The files in Project1 directory are

- P1_image.gif
- P1_logo.gif
- P1_notes.txt
- P1_secret.doc
- P1_secret.html
- P1_secret.txt
- P1_test.html
- project1.html
- secret.doc
- secret.txt
- test.html

Project 1 menu

-

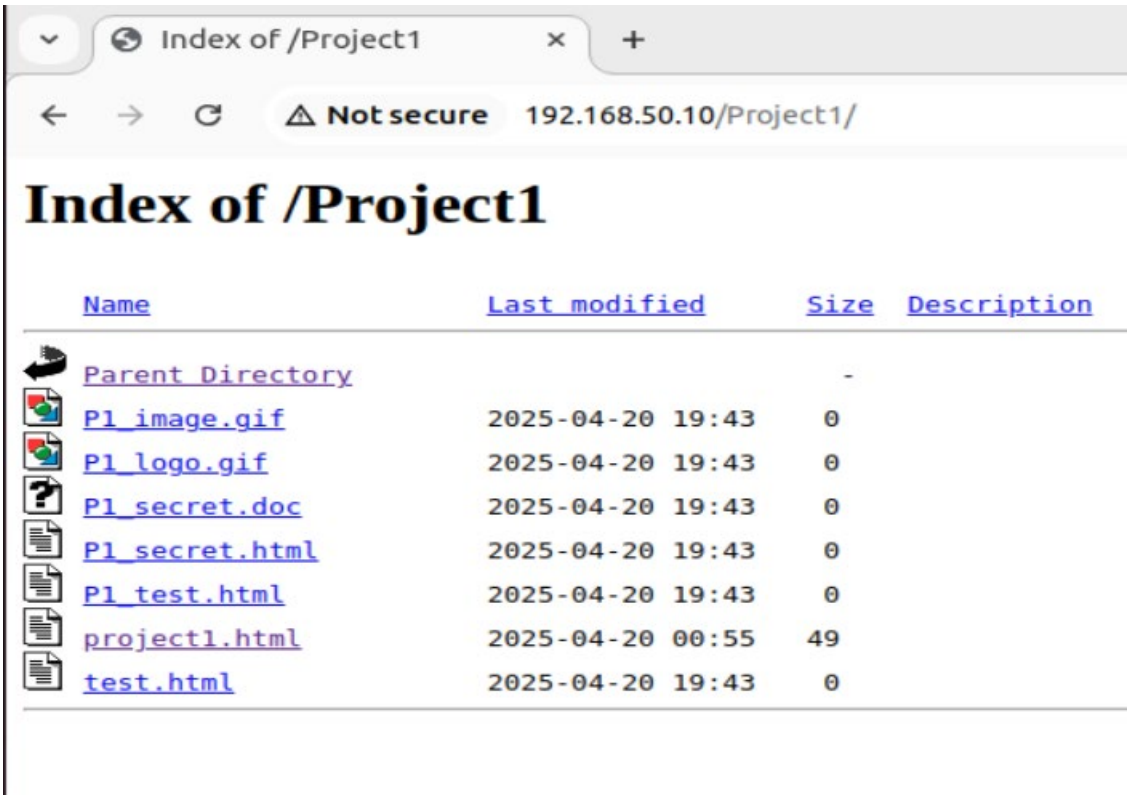
> **Task 3 - Project 1**
>
> - Project1 (192.168.50.10 - All is accessible)
> - Project1 (10.35.16.1 accessible except files secret.* and *.txt)
> - Project1 (10.35.17.1 accessible except files secret.* and *.txt)
> - Project1 (192.168.100.1 Not Accessible)

-

5.3.5.1.1 Project1 (192.168.50.10 - All is accessible)

1. Select First choice in the Menu

2. The following screen appears



Results

A) IP address 192.168.50.10 is used. – “All directories and their contents must be accessible from the 192.168.50.0/24 subnet.”

B) The files :
   - secret.txt
   - secret.doc

   Are not seen as per requirement “ Any files named **secret.*** must not be accessible”
   Note - However, P1_secret.doc and P1_secret.html are displayed or accessible with the current <Files "secret.*"> directive, as it only matches files starting with secret.

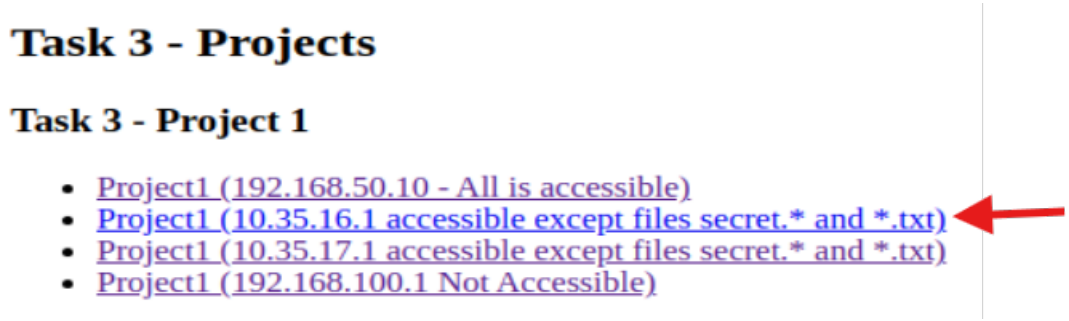
C) The files :

- P1_secret.txt
- P1_notes.txt

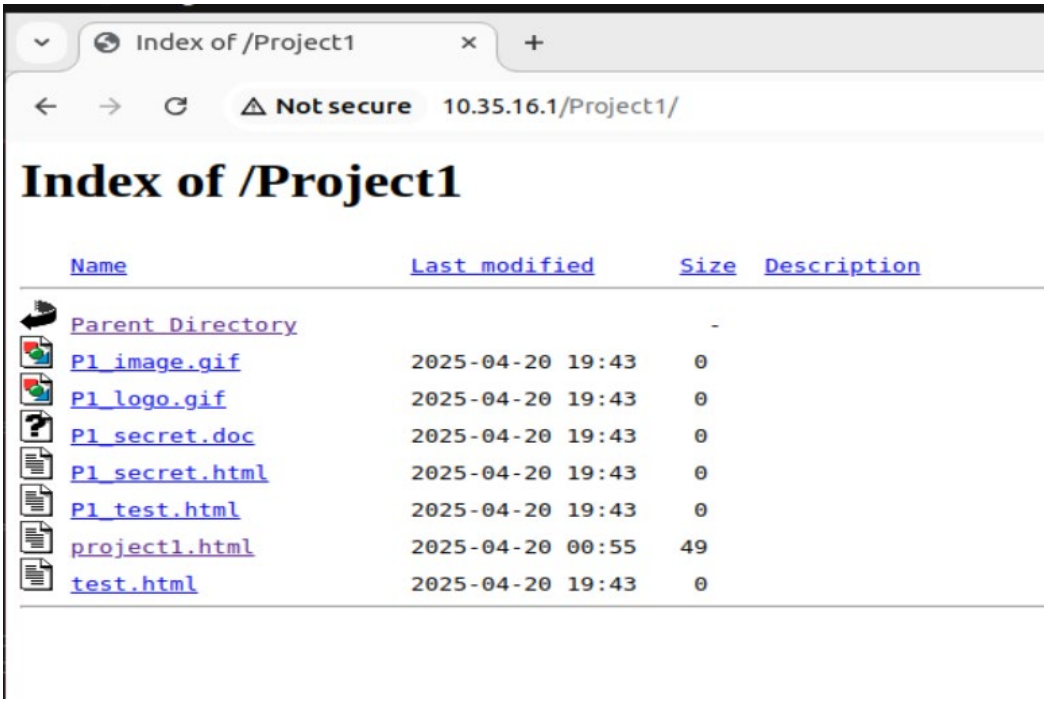
Are not seen as per requirement “For all directories, *.txt files must not be accessible”

5.3.5.1.2 Project1 (10.35.16.1 accessible except files secret.* and *.txt)

1. Select second choice in the Menu



2. The following screen appears



Results

A) IP address 10.35.16.1 is used. – “Accessible only from the 10.35.16.0/24 and 10.35.17.0/24”

B) The files :
   - secret.txt
   - secret.doc

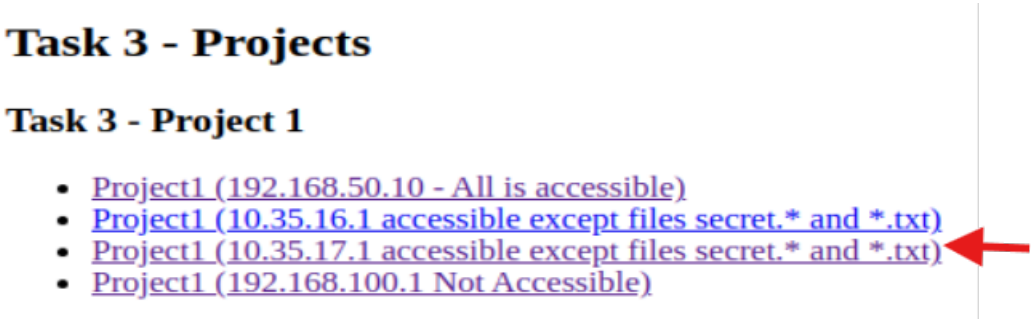
Are not seen as per requirement " Any files named **secret.*** must not be accessible"
Note - However, P1_secret.doc and P1_secret.html are displayed or accessible with the current <Files "secret.*"> directive, as it only matches files starting with secret.

C) The files :
- P1_secret.txt
- P1_notes.txt

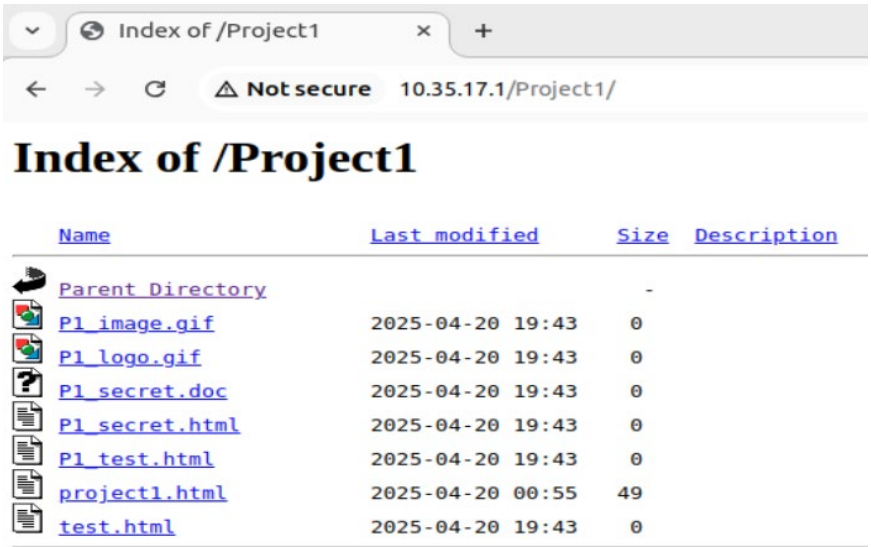
Are not seen as per requirement "For all directories, *.txt files must not be accessible"

5.3.5.1.3 Project1 (10.35.17.1 accessible except files secret.* and *.txt)

1. Select third choice in the menu



2. The following screen appears



Result:

A) IP address 10.35.17.1 is used. – “Accessible only from the 10.35.16.0/24 and 10.35.17.0/24 subnets.”

B) The files :

- secret.txt
- secret.doc

Are not seen as per requirement “ Any files named **secret.*** must not be accessible”
Note - However, P1_secret.doc and P1_secret.html are displayed or accessible with the current <Files "secret.*"> directive, as it only matches files starting with secret.

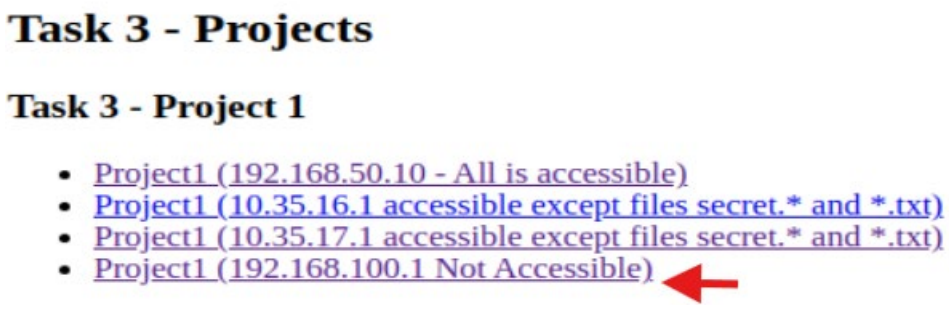
C) The files :

- P1_secret.txt
- P1_notes.txt

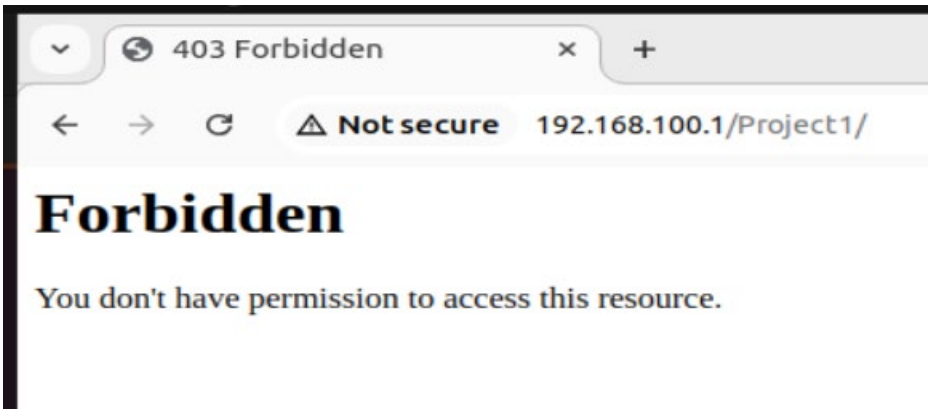
Are not seen as per requirement “For all directories, *.txt files must not be accessible”

5.3.5.1.4 Project1 (192.168.100.1 Not Accessible)

1. Select fourth choice in the menu



2. The following screen appears



Result

192.168.100.0/24 (including 192.168.100.1) is not included in allowed networks, so all requests from this subnet to Project1 should return 403 Forbidden.

5.3.5.2 Project2 testing

Requirement related to Project 2 are tested

1. In the/var/www/html_project1 directory add the following subdirectories: Project2
   ll /var/www/html_project1

```
[root@server1 Documents]# ll /var/www/html_project1
total 56
-rw-r--r--. 1 root   root   26250 Apr 17 12:09 example.jpg
-rw-r--r--. 1 apache apache  4158 Apr 20 14:43 index.html
-rw-r--r--. 1 root   root    2163 Apr 19 03:15 index.html_bkp
drwxr-xr-x. 2 apache apache  4096 Apr 20 19:43 Project1
drwxr-xr-x. 2 apache apache  4096 Apr 20 19:43 Project2
drwxr-xr-x. 2 apache apache  4096 Apr 20 19:43 Project3
drwxr-xr-x. 2 apache apache  4096 Apr 20 19:43 Project4
drwxr-xr-x. 2 apache apache    24 Apr 18 23:57 secure1
drwxr-xr-x. 2 apache apache    24 Apr 18 21:29 secure2
drwxr-xr-x. 2 apache apache    24 Apr 18 21:57 secure3
drwxr-xr-x. 2 apache apache    24 Apr 19 00:07 secure4
drwxr-xr-x. 2 apache apache    41 Apr 19 01:05 secure5
drwxr-xr-x. 2 root   root      41 Apr 19 01:58 secure6
drwxr-xr-x. 2 root   root      41 Apr 19 02:37 secure7
[root@server1 Documents]#
```

2. In each of these directories, create a web page named after the directory itself. For example, Project1/project1.html

   ll /var/www/html_project1/Project2/p*.html

```
[root@server1 Documents]# ll /var/www/html_project1/Project2/p*.html
-rw-r--r--. 1 apache apache 49 Apr 20 00:56 /var/www/html_project1/Project2/project2.html
[root@server1 Documents]#
```

3. Requirements related to httpd configuration

```
# Project2 Configuration
<Directory "/var/www/html_project1/Project2">
    Options +Indexes
    IndexOptions FancyIndexing
    AllowOverride None
    <RequireAll>
        Require all granted
        Require not ip 10.35.16.0/24
    </RequireAll>
</Directory>
```

Directive Explanation:

<Directory "/var/www/html_project1/Project2">

This block specifies configuration rules for the `Project2` directory located at `/var/www/html_project1/Project2`. All settings within this block apply exclusively to this directory and its contents.

Options +Indexes

- Purpose: Enables directory indexing.
- Effect: If a user accesses the directory (e.g., `http://server/Project2/`) without specifying a file like `project2.html`, Apache generates a directory listing showing all files and folders within it.

IndexOptions FancyIndexing

- Purpose: Enhances the appearance of the directory listing enabled by `Options +Indexes`.
- Effect: Adds visual improvements such as:
    - Icons indicating file types.
    - File sorting by name, size, date, or type.
    - A cleaner and more user-friendly display.

AllowOverride None

- Purpose: Disables the use of `.htaccess` files in the `Project2` directory.

- Effect:
    - Any `.htaccess` files present in this directory are ignored by Apache.
    - Ensures that configuration settings can only be modified in the main Apache configuration file, promoting centralized control and security.

<RequireAll>

This block defines access control using "AND" logic, meaning all conditions inside must be met for access to be allowed.

##Inside `<RequireAll>`:

- `Require all granted`:

  Grants access to all users, regardless of IP address.

- `Require not ip 10.35.16.0/24`:

  Specifically denies access to users from the subnet `10.35.16.0/24`.

Summary of Behavior

1. Users can view and browse the directory listing (`Options +Indexes` with `FancyIndexing`).
2. Apache ignores `.htaccess` files in this directory.
3. All users can access the directory except those from the IP range `10.35.16.0/24`, ensuring specific restrictions based on IP.
4. For all directories, *.txt files must not be accessible

```
# Global restriction for all .txt files across all directories
<Files "*.txt">
    Require all denied
</Files>
```

Project 2 files

tree /var/www/html_project1/Project2/

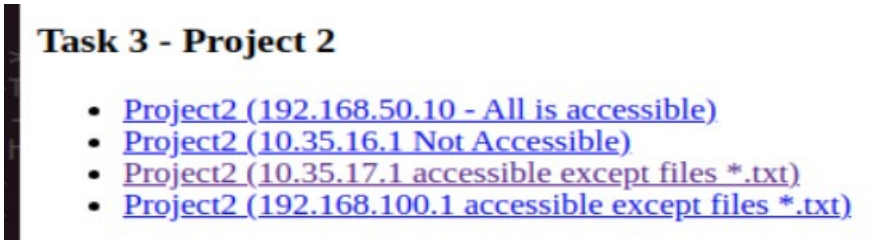
```
[root@server1 Documents]# tree /var/www/html_project1/Project2/
/var/www/html_project1/Project2/
├── P2_image.gif
├── P2_logo.gif
├── P2_notes.txt
├── P2_secret.doc
├── P2_secret.html
├── P2_secret.txt
├── P2_test.html
├── project2.html
├── secret.doc
├── secret.txt
└── test.html

0 directories, 11 files
[root@server1 Documents]#
```

The files in Project2 directory are

- P2_image.gif
- P2_logo.gif
- P2_notes.txt

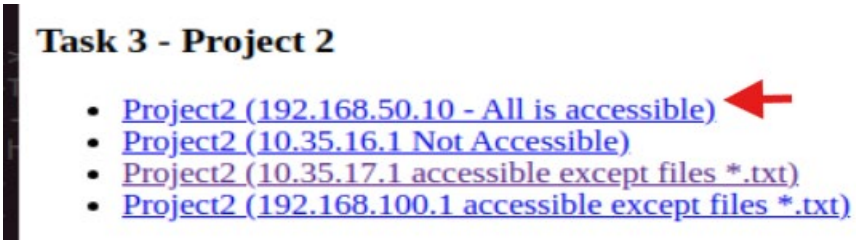
- P2_secret.doc
- P2_secret.html
- P2_secret.txt
- P2_test.html
- Project2.html
- secret.doc
- secret.txt
- test.html

Project 2 menu

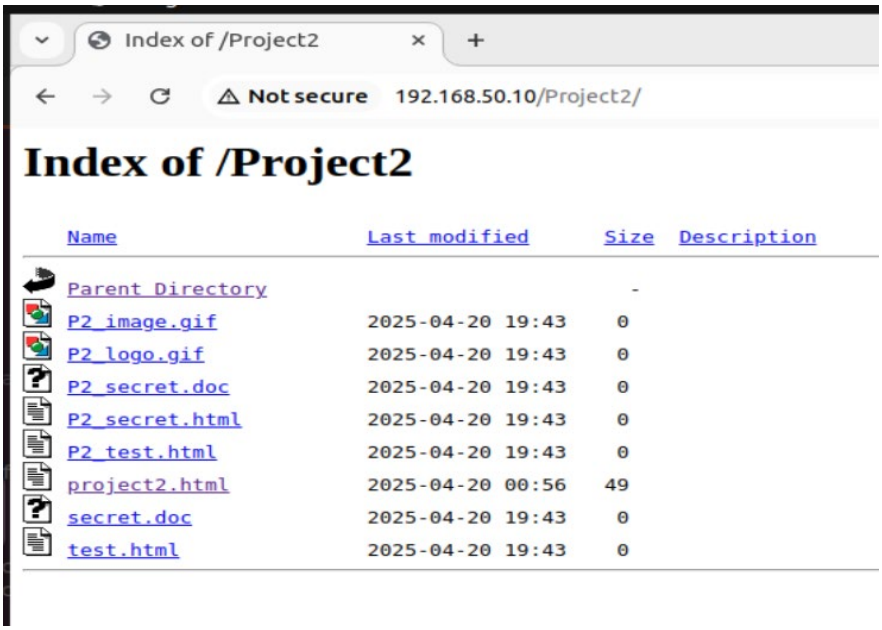


5.3.5.2.1 Project2 (192.168.50.10 - All is accessible)

1. Select first choice in menu



2. The following screen will appear



Results

A) IP address 192.168.50.10 is used. – “All directories and their contents must be accessible from the 192.168.50.0/24 subnet.”

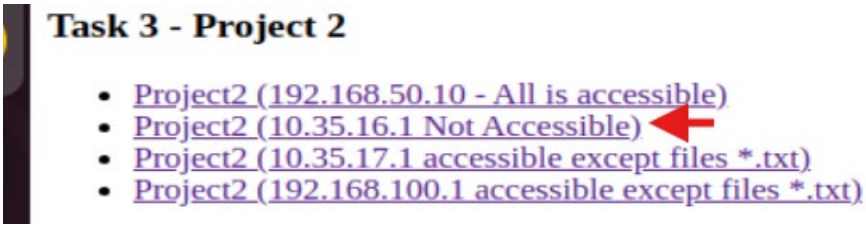
B) The files :

- secret.txt
- P1_secret.txt
- P1_notes.txt

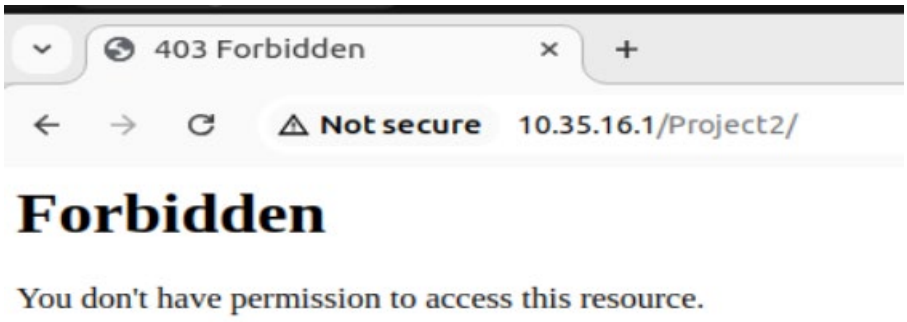
Are not seen as per requirement “For all directories, *.txt files must not be accessible”

5.3.5.2.2 Project2 (10.35.16.1 - Not Accessible)

1. Select second choice in menu



2. The following screen will appear



Results

A) IP Address 10.35.16.1 - Not accessible only from the 10.35.16.0/24 subnet. All files must be accessible to others. IP Address 10.35.16.1 is NOT allowed.

5.3.5.2.3 Project2 (10.35.17.1 - accessible except files *.txt)

1. Select third choice in menu

2. The following screen will appear

Results

A) IP Address 10.35.17.1 - Not accessible only from the 10.35.16.0/24 subnet. All files must be accessible to others. IP Address 10.35.17.1 is allowed

B) The files :
   - secret.txt
   - P1_secret.txt
   - P1_notes.txt

   Are not seen as per requirement "For all directories, *.txt files must not be accessible"

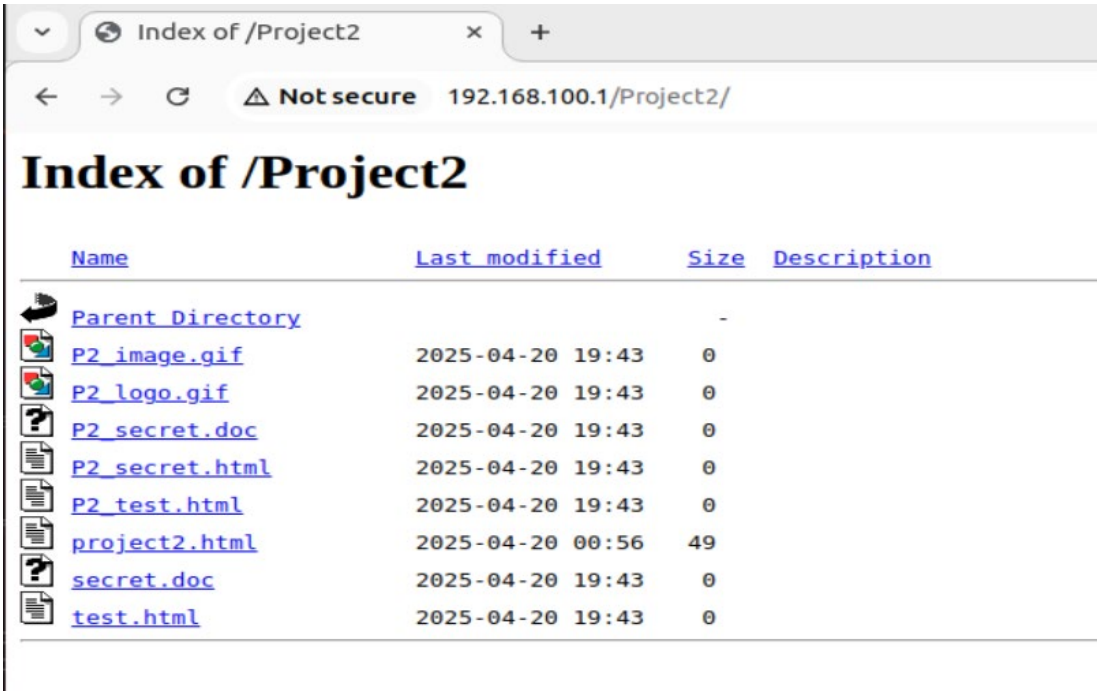
5.3.5.2.4 Project2 (192.168.100.1 - accessible except files *.txt)

1. Select fourth choice in menu

Task 3 - Project 2

- Project2 (192.168.50.10 - All is accessible)
- Project2 (10.35.16.1 Not Accessible)
- Project2 (10.35.17.1 accessible except files *.txt)
- Project2 (192.168.100.1 accessible except files *.txt)

2. The following screen will appear



Results

A) IP Address 192.168.100.1 - Not accessible only from the 10.35.16.0/24 subnet. All files must be accessible to others. IP Address 192.168.100.1 is allowed

B) The files :

- secret.txt
- P1_secret.txt
- P1_notes.txt

Are not seen as per requirement "For all directories, *.txt files must not be accessible"

5.3.5.3 Project3 testing

Requirement related to Project 3 are tested

1. In the/var/www/html_project1 directory add the following subdirectories: Project3
   ll /var/www/html_project1

```
[root@server1 Documents]# ll /var/www/html_project1
total 56
-rw-r--r--. 1 root   root   26250 Apr 17 12:09 example.jpg
-rw-r--r--. 1 apache apache  4158 Apr 20 14:43 index.html
-rw-r--r--. 1 root   root    2163 Apr 19 03:15 index.html_bkp
drwxr-xr-x. 2 apache apache  4096 Apr 20 19:43 Project1
drwxr-xr-x. 2 apache apache  4096 Apr 20 19:43 Project2
drwxr-xr-x. 2 apache apache  4096 Apr 20 19:43 Project3
drwxr-xr-x. 2 apache apache  4096 Apr 20 19:43 Project4
drwxr-xr-x. 2 apache apache    24 Apr 18 23:57 secure1
drwxr-xr-x. 2 apache apache    24 Apr 18 21:29 secure2
drwxr-xr-x. 2 apache apache    24 Apr 18 21:57 secure3
drwxr-xr-x. 2 apache apache    24 Apr 19 00:07 secure4
drwxr-xr-x. 2 apache apache    41 Apr 19 01:05 secure5
drwxr-xr-x. 2 root   root      41 Apr 19 01:58 secure6
drwxr-xr-x. 2 root   root      41 Apr 19 02:37 secure7
[root@server1 Documents]#
```

2. In each of these directories, create a web page named after the directory itself. For example, Project3/project3.html

   ll /var/www/html_project1/Project3/p*.html

```
[root@server1 Documents]# ll /var/www/html_project1/Project3/p*.html
-rw-r--r--. 1 apache apache 49 Apr 20 00:57 /var/www/html_project1/Project3/project3.html
[root@server1 Documents]#
```

3. Requirements related to httpd configuration

```
# Project3 Configuration
<Directory "/var/www/html_project1/Project3">
    Options +Indexes
    IndexOptions FancyIndexing
    AllowOverride None
    <RequireAny>
        Require ip 192.168.100.0/24
        Require ip 192.168.50.0/24
    </RequireAny>
    <Files "*.gif">
        Require all denied
    </Files>
</Directory>
```

Directive Explanation:

<Directory "/var/www/html_project1/Project3">

This block applies rules specifically to the directory `/var/www/html_project1/Project3` and its contents. Any configuration inside this block only affects this directory.

Options +Indexes

- Purpose: Enables directory listing.
- Effect: If a user accesses `http://server/Project3/` without specifying a file (e.g., `project3.html`), Apache will generate and display an index of all files within the directory.

IndexOptions FancyIndexing

- Purpose: Beautifies the directory index created by `Options +Indexes`.
- Effect: Enhances the directory listing with features such as:
  - Icons for different file types.
  - Options to sort files by size, type, or last modified date.
  - A more user-friendly layout for browsing.

AllowOverride None

- Purpose: Disables `.htaccess` files within the directory.
- Effect: Prevents any settings in `.htaccess` from altering the directory's behavior. All configurations for `/Project3` must be defined in the main Apache configuration file for centralized control and security.

<RequireAny>

This block allows access to the directory if any one of the conditions specified inside it is satisfied.

Conditions in `<RequireAny>`:

1. `Require ip 192.168.100.0/24`:
   - Allows access for users from the subnet `192.168.100.0/24`.
2. `Require ip 192.168.50.0/24`:
   - Allows access for users from the subnet `192.168.50.0/24`.

Combined Effect:

Access is granted to users from either `192.168.100.0/24` or `192.168.50.0/24`.

<Files "*.gif">

This block defines specific rules for files matching the pattern `*.gif` within `/Project3`.

Inside `<Files "*.gif">`:

- `Require all denied`:

- Denies access to all `.gif` files, regardless of the user's IP or any other conditions.

Behavior Summary

1. Directory Listing:
   - When users access `/Project3/`, Apache will display an enhanced directory listing (`+Indexes` with `FancyIndexing`).

2. Access Restrictions:
   - Only users from the subnets `192.168.100.0/24` and `192.168.50.0/24` are allowed to access the directory.

3. File-Specific Restriction:
   - All `.gif` files (e.g., `image.gif`, `logo.gif`) in `/Project3` are explicitly blocked, regardless of the user's IP.

4. Centralized Configuration:
   - The `AllowOverride None` directive ensures that no `.htaccess` files in `/Project3` can modify or override these rules.

5. For all directories, *.txt files must not be accessible

```
# Global restriction for all .txt files across all directories
<Files "*.txt">
    Require all denied
</Files>
```

Project 3 files

tree /var/www/html_project1/Project3/

```
[root@server1 Documents]# tree /var/www/html_project1/Project3/
/var/www/html_project1/Project3/
├── P3_image.gif
├── P3_logo.gif
├── P3_notes.txt
├── P3_secret.doc
├── P3_secret.html
├── P3_secret.txt
├── P3_test.html
├── project3.html
├── secret.doc
├── secret.txt
└── test.html

0 directories, 11 files
[root@server1 Documents]#
```

The files in Project3 directory are

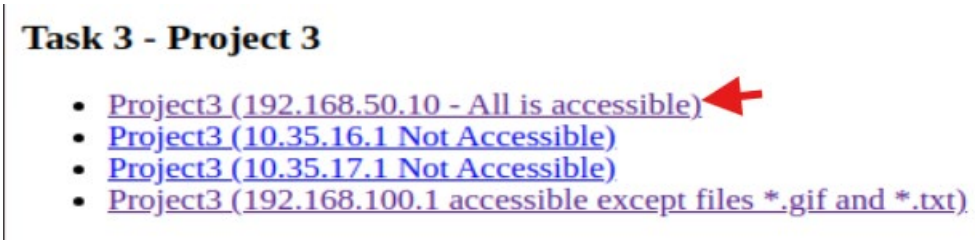
- P3_image.gif
- P3_logo.gif
- P3_notes.txt
- P3_secret.doc
- P3_secret.html
- P3_secret.txt
- P3_test.html
- project3.html
- secret.doc
- secret.txt
- test.html

Project 3 menu

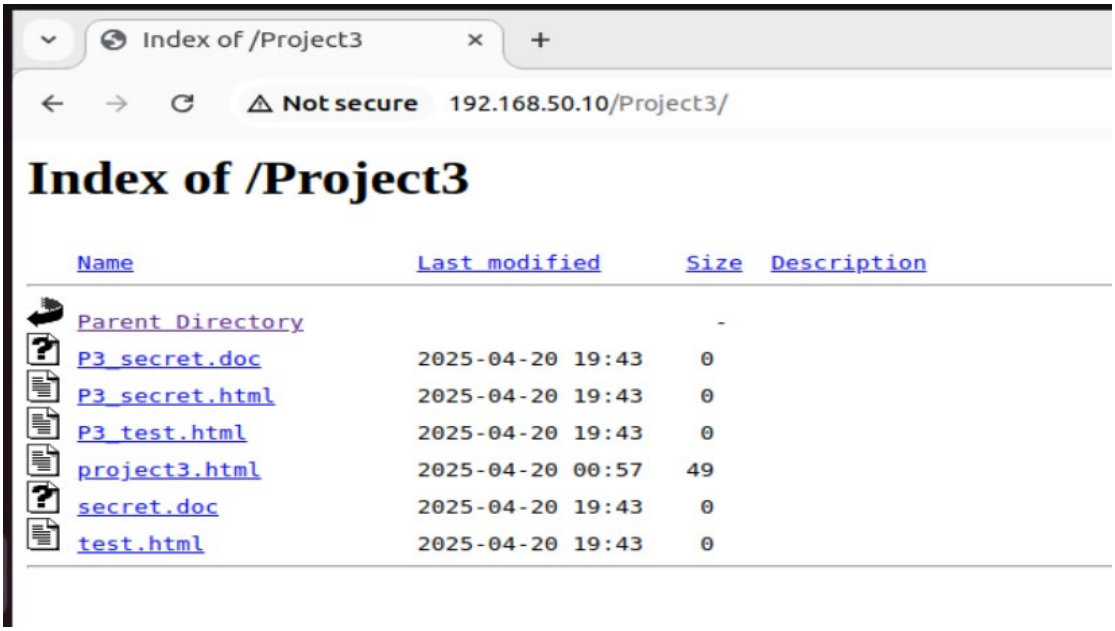
> **Task 3 - Project 3**
>
> - Project3 (192.168.50.10 - All is accessible)
> - Project3 (10.35.16.1 Not Accessible)
> - Project3 (10.35.17.1 Not Accessible)
> - Project3 (192.168.100.1 accessible except files *.gif and *.txt)

5.3.5.3.1 Project3 (192.168.50.10 - All is accessible)

1. Select first choice in menu



2. The following screen will appear



Results

A) IP address 192.168.50.10 is used. – "All directories and their contents must be accessible from the 192.168.50.0/24 subnet."

B) The files :
   - secret.txt
   - P3_secret.txt
   - P3_notes.txt

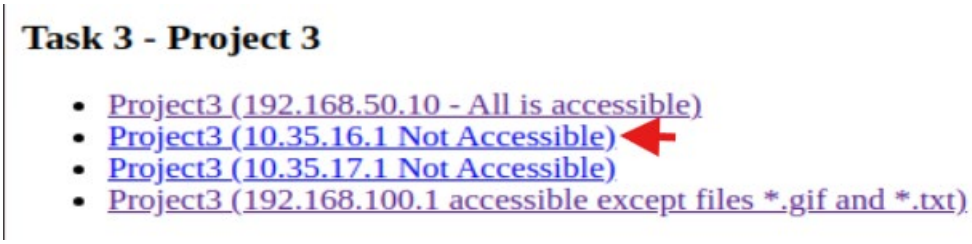
   Are not seen as per requirement "For all directories, *.txt files must not be accessible"

C) The files:
   - P3_image.gif
   - P3_logo.gif

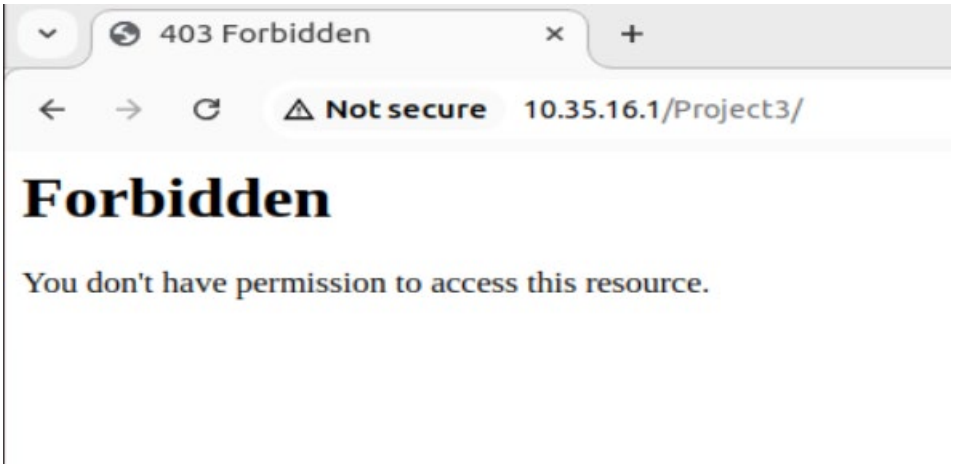
   Are not seen as per requirement "All *.gif files must not be accessible."

5.3.5.3.2 Project3 (10.35.16.1 - Not Accessible)

1. Select second choice in menu



2. The following screen will appear

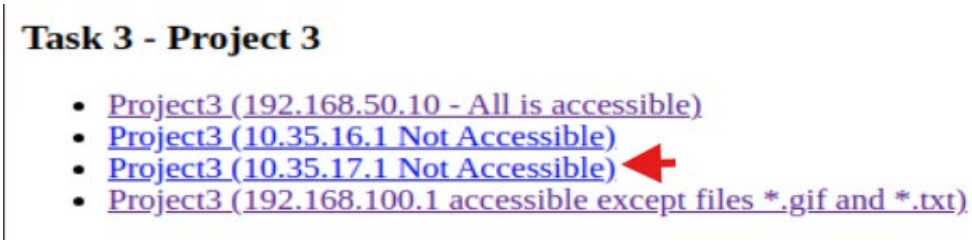


Results

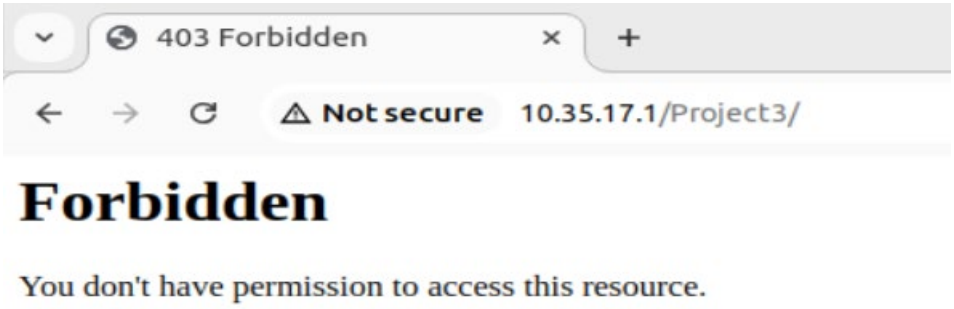
A) IP Address 10.35.16.1 - Accessible only from the 192.168.100.0/24 subnet . IP Address 10.35.16.1 is NOT allowed.

5.3.5.3.3 Project3 (10.35.17.1 - Not Accessible)

1. Select third choice in menu



2. The following screen will appear

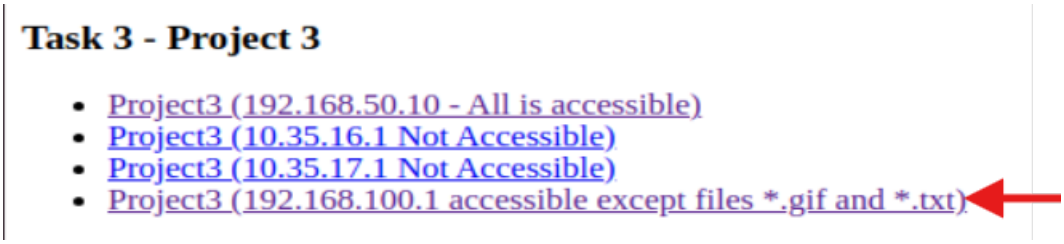


Results

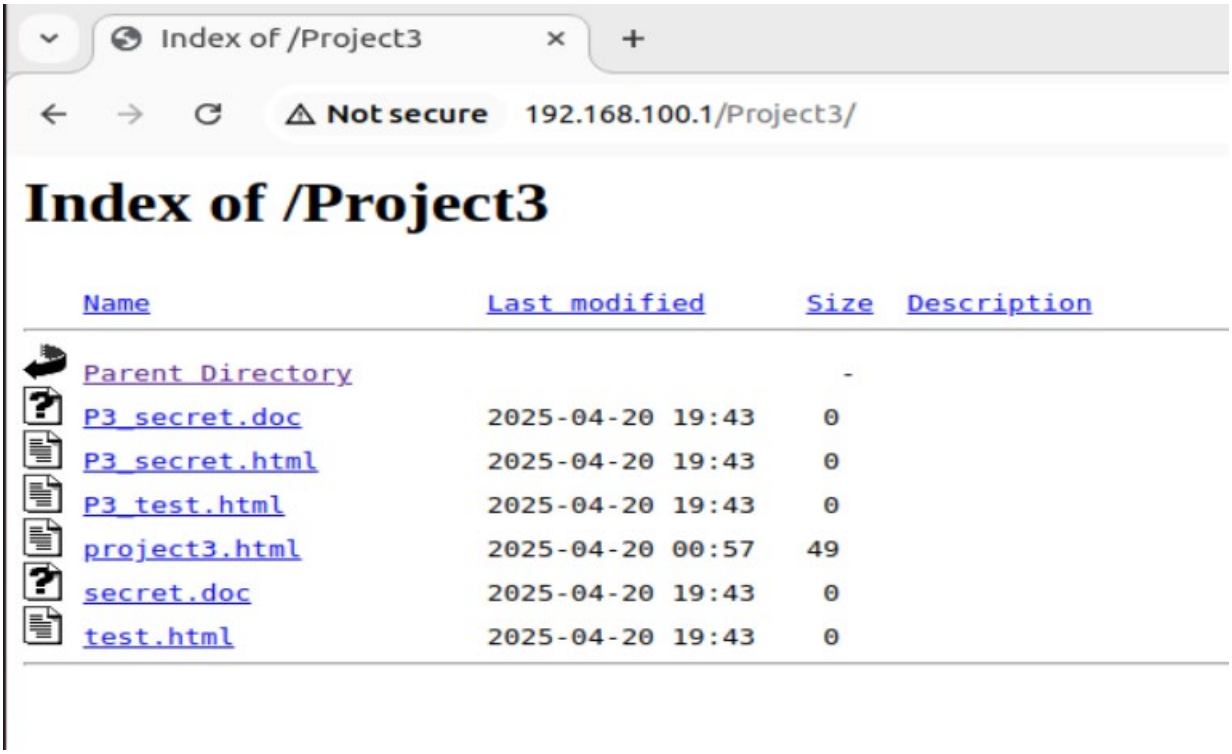
A) IP Address 10.35.17.1 - Accessible only from the 192.168.100.0/24 subnet . IP Address 10.35.17.1 is NOT allowed.

5.3.5.3.4 Project3 (192.168.100.1 - accessible except files *.gif and *.txt)

1. Select fourth choice in menu



2. The following screen will appear



Results

A) IP Address 192.168.100.1 - Not accessible only from the 10.35.16.0/24 subnet. All files must be accessible to others. IP Address 192.168.100.1 is allowed

B) The files :

- secret.txt
- P3_secret.txt
- P3_notes.txt

Are not seen as per requirement "For all directories, *.txt files must not be accessible"

C) The files:
- P3_image.gif
- P3_logo.gif

Are not seen as per requirement "All *.gif files must not be accessible."

5.3.5.4 Project4 testing

Requirement related to Project 3 are tested

1. In the/var/www/html_project1 directory add the following subdirectories: Project4

   ll /var/www/html_project1

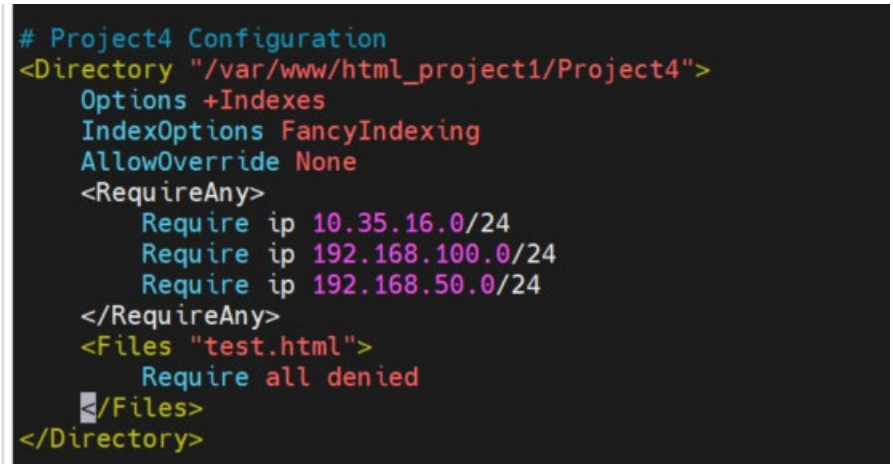
```
[root@server1 Documents]# ll /var/www/html_project1
total 56
-rw-r--r--. 1 root   root   26250 Apr 17 12:09 example.jpg
-rw-r--r--. 1 apache apache  4158 Apr 20 14:43 index.html
-rw-r--r--. 1 root   root    2163 Apr 19 03:15 index.html_bkp
drwxr-xr-x. 2 apache apache  4096 Apr 20 19:43 Project1
drwxr-xr-x. 2 apache apache  4096 Apr 20 19:43 Project2
drwxr-xr-x. 2 apache apache  4096 Apr 20 19:43 Project3
drwxr-xr-x. 2 apache apache  4096 Apr 20 19:43 Project4
drwxr-xr-x. 2 apache apache    24 Apr 18 23:57 secure1
drwxr-xr-x. 2 apache apache    24 Apr 18 21:29 secure2
drwxr-xr-x. 2 apache apache    24 Apr 18 21:57 secure3
drwxr-xr-x. 2 apache apache    24 Apr 19 00:07 secure4
drwxr-xr-x. 2 apache apache    41 Apr 19 01:05 secure5
drwxr-xr-x. 2 root   root      41 Apr 19 01:58 secure6
drwxr-xr-x. 2 root   root      41 Apr 19 02:37 secure7
[root@server1 Documents]#
```

2. In each of these directories, create a web page named after the directory itself. For example, Project4/project4.html

   ll /var/www/html_project1/Project4/p*.html

```
[root@server1 Documents]# ll /var/www/html_project1/Project4/p*.html
-rw-r--r--. 1 apache apache 49 Apr 20 00:56 /var/www/html_project1/Project4/project4.html
[root@server1 Documents]#
```

3. Requirements related to httpd configuration



```
# Project4 Configuration
<Directory "/var/www/html_project1/Project4">
    Options +Indexes
    IndexOptions FancyIndexing
    AllowOverride None
    <RequireAny>
        Require ip 10.35.16.0/24
        Require ip 192.168.100.0/24
        Require ip 192.168.50.0/24
    </RequireAny>
    <Files "test.html">
        Require all denied
    </Files>
</Directory>
```

Directive Explanation:

<Directory "/var/www/html_project1/Project4">

This block defines rules and settings that apply to the directory `/var/www/html_project1/Project4` and its contents. All configurations within this block affect the behavior of Apache for this specific directory.

Options +Indexes

- Purpose: Enables directory listing.
- Effect: If a user accesses `http://server/Project4/` without specifying a specific file (e.g., `project4.html`), Apache generates and displays an index of all files in the directory.

IndexOptions FancyIndexing

- Purpose: Beautifies the directory listing created by `Options +Indexes`.
- Effect: Adds features such as:
    - Icons for file types.
    - Options to sort files (by name, size, type, or modification date).
    - An improved, user-friendly directory listing appearance.

AllowOverride None

- Purpose: Prevents the use of `.htaccess` files in the `/Project4` directory.
- Effect: Ensures all configuration changes for this directory must be made directly in the main Apache configuration file, centralizing control and improving security.

<RequireAny>

This block uses "OR" logic for access control, meaning access to the directory is allowed if any one of the listed conditions is met.

Inside `**<RequireAny>**`:

- `Require ip 10.35.16.0/24`:
    - Grants access to users from the subnet `10.35.16.0/24`.
- `Require ip 192.168.100.0/24`:

    - Grants access to users from the subnet `192.168.100.0/24`.
- `Require ip 192.168.50.0/24`:
    - Grants access to users from the subnet `192.168.50.0/24`.

Combined Effect: Users from any of these subnets can access the directory.

<Files "test.html">

This block applies specific access control rules to any file named `test.html` within the `/Project4` directory.

Inside `<Files "test.html">`:

- `Require all denied`:
    - Denies access to the file `test.html` for all users, regardless of IP address or any other condition.

Effect:

Even though directory access is granted to specific IP ranges, the file `test.html` is always blocked from being accessed.

Behavior Summary

1. Directory Listing:
   When users access `/Project4/`, Apache provides a directory listing styled with `FancyIndexing`, allowing users to see the files in the folder.

2. Access Restrictions:
   Only users from the subnets `10.35.16.0/24`, `192.168.100.0/24`, and `192.168.50.0/24` can access the directory and its files.

3. File-Specific Restriction:
   The file `test.html` is explicitly denied access to all users, regardless of their IP address.

4. Security:
   Disabling `.htaccess` files ensures configurations cannot be overridden locally within the directory.

5. For all directories, *.txt files must not be accessible

```
# Global restriction for all .txt files across all directories
<Files "*.txt">
    Require all denied
</Files>
```

Project 4 files

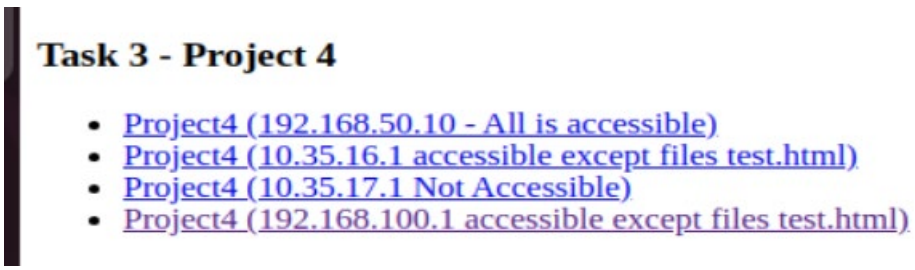
tree /var/www/html_project1/Project4/

```
[root@server1 Documents]# tree /var/www/html_project1/Project4/
/var/www/html_project1/Project4/
├── P4_image.gif
├── P4_logo.gif
├── P4_notes.txt
├── P4_secret.doc
├── P4_secret.html
├── P4_secret.txt
├── P4_test.html
├── project4.html
├── secret.doc
├── secret.txt
└── test.html

0 directories, 11 files
[root@server1 Documents]#
```

The files in Project4 directory are

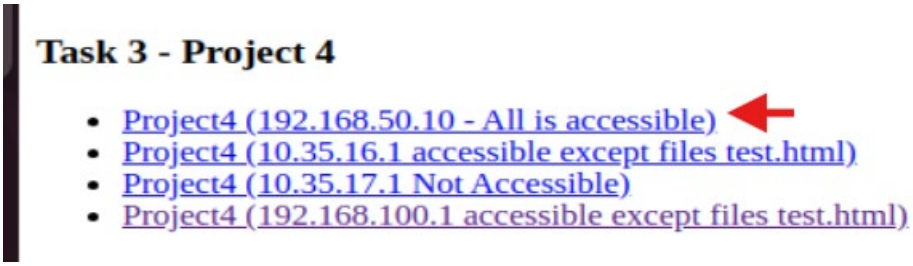
- P4_image.gif
- P4_logo.gif
- P4_notes.txt
- P4_secret.doc
- P4_secret.html
- P4_secret.txt
- P4_test.html
- project4.html
- secret.doc
- secret.txt
- test.html

Project 4 menu

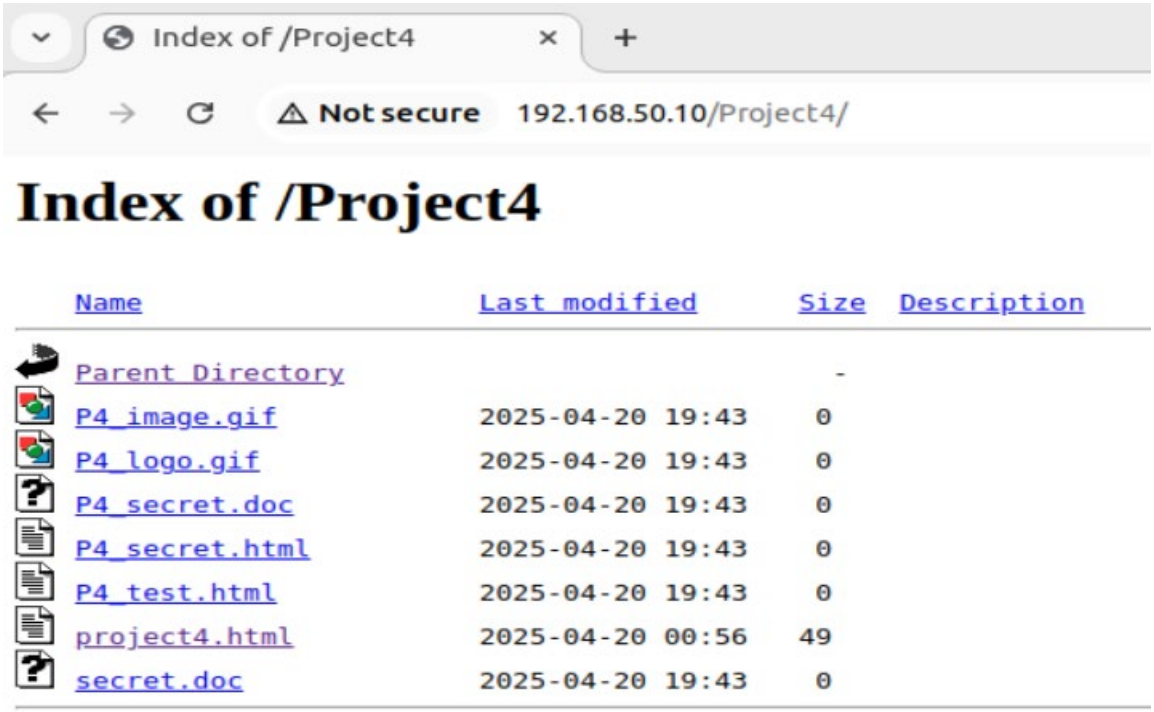


5.3.5.4.1 Project4 (192.168.50.10 - All is accessible)

1. Select first choice in menu



2. The following screen will appear



Results

A) IP address 192.168.50.10 is used. – "All directories and their contents must be accessible from the 192.168.50.0/24 subnet."

B) The files :
   - secret.txt
   - P3_secret.txt
   - P3_notes.txt

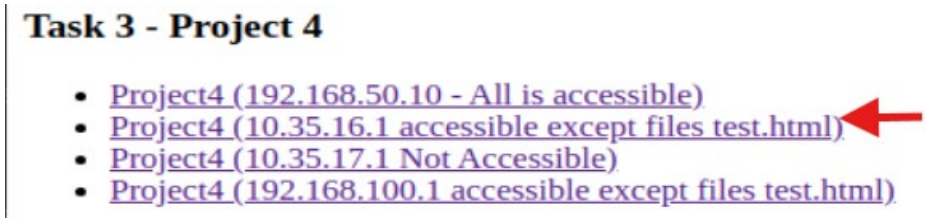
   Are not seen as per requirement "For all directories, *.txt files must not be accessible"

C) The files:
   - test.html

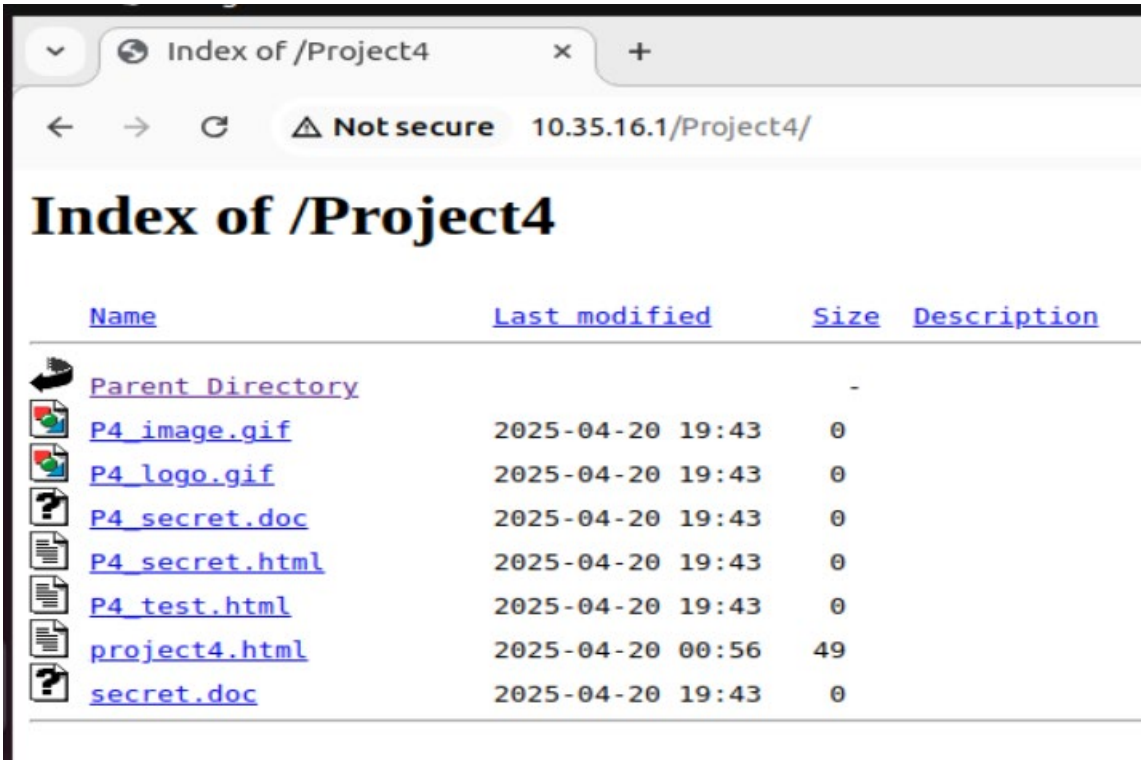
   Is not seen as per requirement "All test.html files must not be accessible.

5.3.5.4.2 Project4 (10.35.16.1 - Not Accessible)

1. Select second choice in menu



2. The following screen will appear



Results

A) IP Address 10.35.16.1 - Accessible only from 10.35.16.0/24 and 192.168.100.0/24. IP Address 10.35.16.1 is allowed.

B) The files :
   - secret.txt
   - P4_secret.txt
   - P4_notes.txt

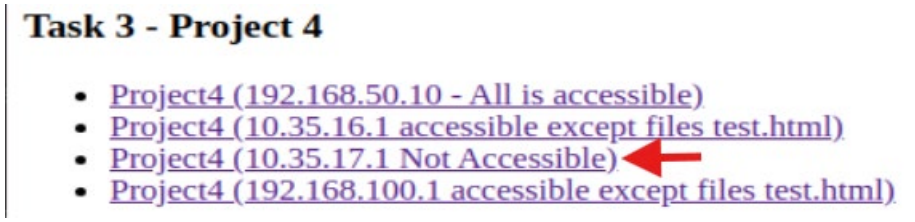
   Are not seen as per requirement "For all directories, *.txt files must not be accessible"

C) The files:
   - test.html

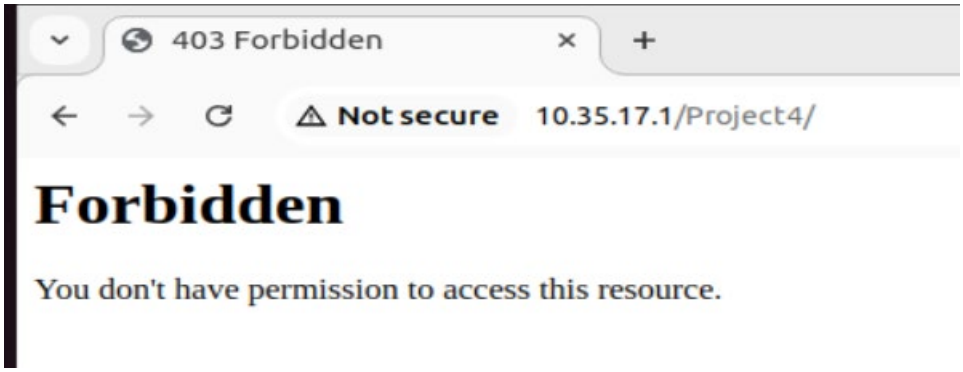
   Is not seen as per requirement "All test.html files must not be accessible.

5.3.5.4.3 Project4 (10.35.17.1 - Not Accessible)

1. Select third choice in menu



2. The following screen will appear

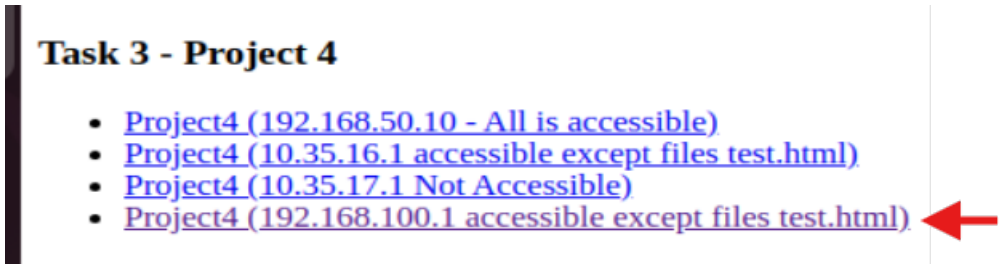


Results

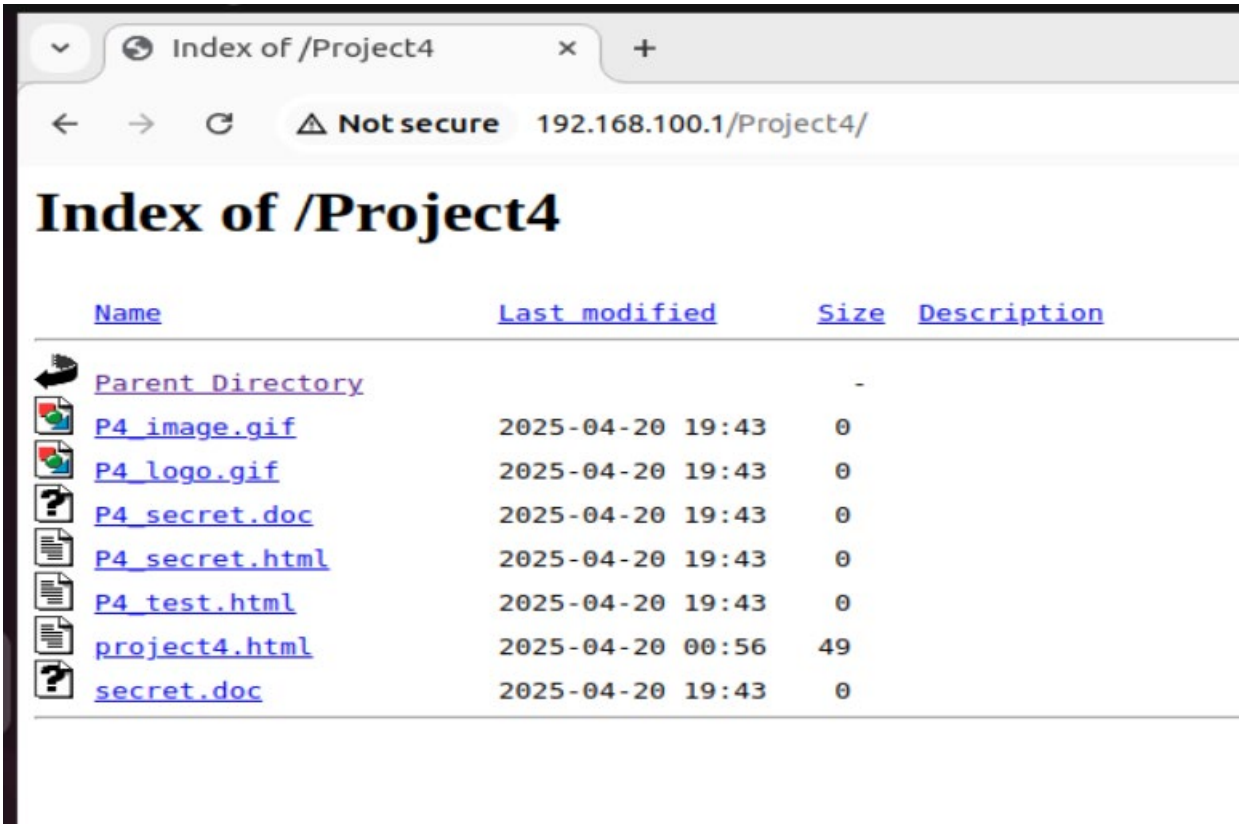
A) IP Address 10.35.17.1 - Accessible only from 10.35.16.0/24 and 192.168.100.0/24. IP Address 10.35.17.1 is NOT allowed.

5.3.5.4.4 Project4 (192.168.100.1 - accessible except files test.html)

1. Select fourth choice in menu



2. The following screen will appear



Results

A) IP Address 192.168.100.1 - Accessible only from 10.35.16.0/24 and 192.168.100.0/24. IP Address 192.168.100.1 is allowed.

B) The files :
   - secret.txt
   - P4_secret.txt
   - P4_notes.txt

   Are not seen as per requirement "For all directories, *.txt files must not be accessible"

C) The files:
   - test.html

   Is not seen as per requirement "All test.html files must not be accessible.

6 TASK 4 – AUTHORIZATION

6.1 Requirements

1. Identify the network subnet of each department:
   - **Vendors** use the subnet **10.50.1.0/24**
   - **Accountants** use the subnet **10.51.1.0/24**
   - **Administrators** use the subnet **10.52.1.0/24**
   - **Programmers** use the subnet **10.53.1.0/24**
2. Each department has its own **dedicated web directory** on the server.
3. Websites must be placed in the **/var/www/htdocs/*<group_name>*** directory. For example: **/var/www/htdocs/vendors**, **/var/www/htdocs/accountants**, etc.
4. You must use **aliases** to make these websites accessible. For example, to access the vendors' website, use: **http://10.50.1.1/vendors**
5. Configure the Apache server with the following access rules:
   - Vendors can access only their own website.
   - Accountants can access only the accountants' website and must not be able to view any *.html files.
   - Administrators must be able to view all department websites.
   - Programmers must be able to access their own website and to view all department websites except their *.gif or *.jpg files.

Note: Make sure to demonstrate the enforcement of each access control rule through appropriate tests and screenshots or logs as evidence.

6.2 Modify Main html

Modify main html file to include Task 3 menu

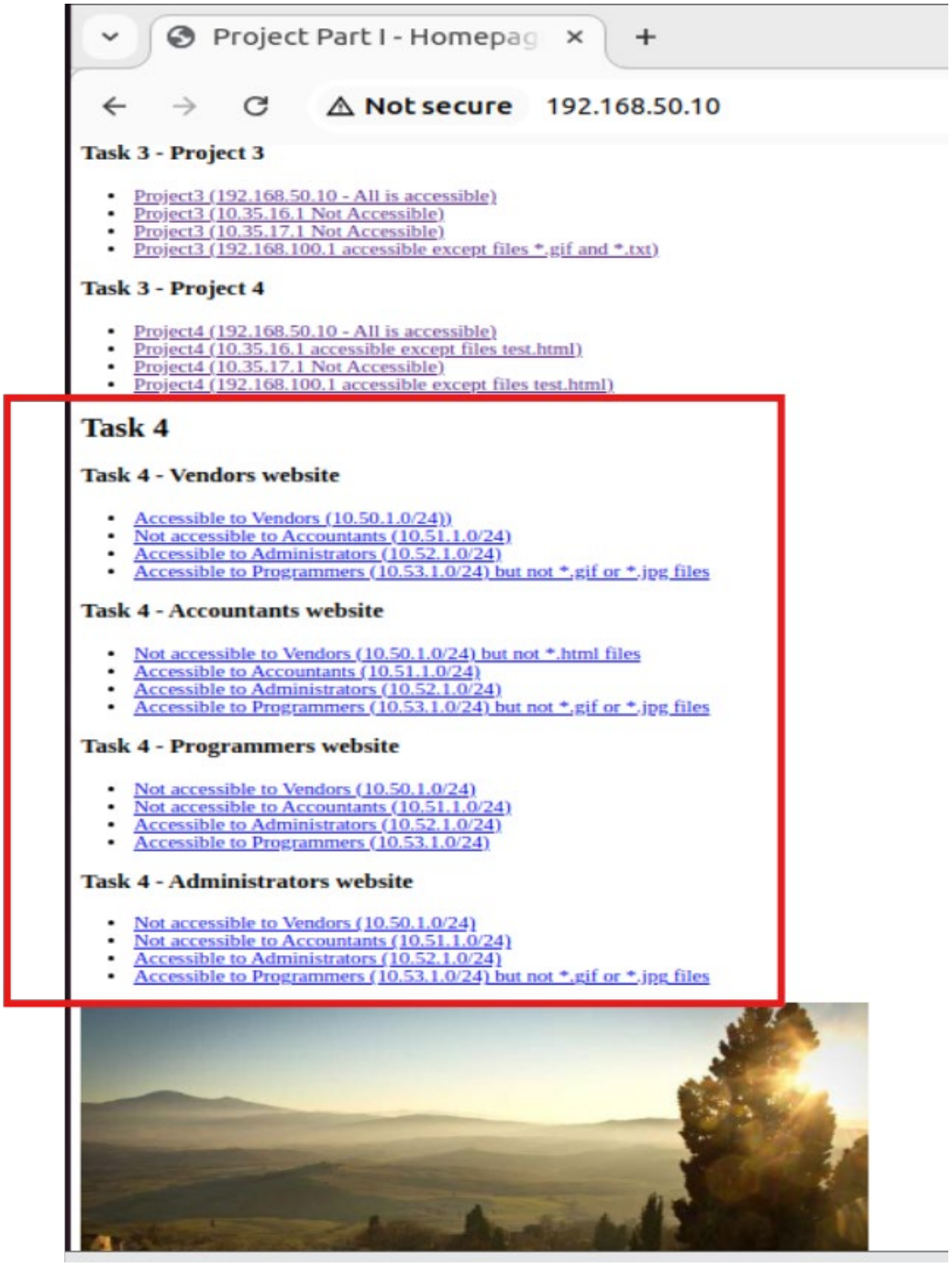
Open the file for edition and include the lines in the box, related to Task 4

vi /var/www/html_project1/index.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Project Part I - Homepage</title>
</head>
<body>
    <p><b><i><u>Welcome to Project Part I</u></i></b></p>
    <hr>

    <!-- Task 2 - Secure Directories -->
    <h2>Task 2 - Secure Directories</h2>
    <ul>
        <li><a href="http://192.168.50.10/secure1/index.html">Secure1 (user01 -
Accessible)</a></li>
        <li><a href="http://192.168.50.10/secure2/index.html">Secure2 (user01 and
192.168.50.0/24 - Accessible)</a></li>
        <li><a href="http://10.35.16.1/secure2/index.html">Secure2 (user01 and 10.35.16.1/24 -
Not Accessible)</a></li>
        <li><a href="http://192.168.50.10/secure3/index.html">Secure3 (user01 or
192.168.50.0/24 - Accessible)</a></li>
        <li><a href="http://10.35.16.1/secure3/index.html">Secure3 (user01 or 10.35.16.1/24 -
Not Accessible)</a></li>
        <li><a href="http://192.168.50.10/secure4/index.html">Secure4 (user02 -
Accessible)</a></li>
        <li><a href="http://192.168.50.10/secure5/index.html">Secure5 (user01 with .htaccess -
Accessible)</a></li>
        <li><a href="http://192.168.50.10/secure6/index.html">Secure6 (user01 and
192.168.50.0/24 with .htaccess - Accessible)</a></li>
        <li><a href="http://10.35.16.1/secure6/index.html">Secure6 (user01 and 10.35.16.1/24
with .htaccess - Not Accessible)</a></li>
        <li><a href="http://192.168.50.10/secure7/index.html">Secure7 (user01 or
192.168.50.0/24 with .htaccess - Accessible)</a></li>
        <li><a href="http://10.35.16.1/secure7/index.html">Secure7 (user01 or 10.35.16.1/24
with .htaccess - Accessible)</a></li>
    </ul>

    <!-- Task 3 - Projects -->
    <h2>Task 3 - Projects</h2>

    <!-- Task 3 - Project 1 -->
    <h3>Task 3 - Project 1</h3>
    <ul>
        <li><a href="http://192.168.50.10/Project1/">Project1 (192.168.50.10 - All is
accessible)</a></li>
        <li><a href="http://10.35.16.1/Project1/">Project1 (10.35.16.1 accessible except files
secret.* and *.txt)</a></li>
        <li><a href="http://10.35.17.1/Project1/">Project1 (10.35.17.1 accessible except files
secret.* and *.txt)</a></li>
        <li><a href="http://192.168.100.1/Project1/">Project1 (192.168.100.1 Not
Accessible)</a></li>
    </ul>

    <!-- Task 3 - Project 2 -->
    <h3>Task 3 - Project 2</h3>
    <ul>
        <li><a href="http://192.168.50.10/Project2/">Project2 (192.168.50.10 - All is
accessible)</a></li>
        <li><a href="http://10.35.16.1/Project2/">Project2 (10.35.16.1 Not Accessible)</a></li>
        <li><a href="http://10.35.17.1/Project2/">Project2 (10.35.17.1 accessible except files
*.txt)</a></li>
        <li><a href="http://192.168.100.1/Project2/">Project2 (192.168.100.1 accessible except
files *.txt)</a></li>
    </ul>

    <!-- Task 3 - Project 3 -->
    <h3>Task 3 - Project 3</h3>
    <ul>
        <li><a href="http://192.168.50.10/Project3/">Project3 (192.168.50.10 - All is
accessible)</a></li>
        <li><a href="http://10.35.16.1/Project3/">Project3 (10.35.16.1 Not Accessible)</a></li>
```

```
        <li><a href="http://10.35.17.1/Project3/">Project3 (10.35.17.1 Not Accessible)</a></li>
        <li><a href="http://192.168.100.1/Project3/">Project3 (192.168.100.1 accessible except
files *.gif and *.txt)</a></li>
    </ul>

    <!-- Task 3 - Project 4 -->
    <h3>Task 3 - Project 4</h3>
    <ul>
        <li><a href="http://192.168.50.10/Project4/">Project4 (192.168.50.10 - All is
accessible)</a></li>
        <li><a href="http://10.35.16.1/Project4/">Project4 (10.35.16.1 accessible except files
test.html)</a></li>
        <li><a href="http://10.35.17.1/Project4/">Project4 (10.35.17.1 Not Accessible)</a></li>
        <li><a href="http://192.168.100.1/Project4/">Project4 (192.168.100.1 accessible except
files test.html)</a></li>
    </ul>


 <!-- Task 4 - -->
    <h2>Task 4</h2>

    <!-- Task 4 - Vendors website -->
    <h3>Task 4 - Vendors website</h3>
    <ul>
        <li><a href="http://10.50.1.1/vendors">Accessible to Vendors (10.50.1.0/24))</a></li>
        <li><a href="http://10.51.1.1/vendors">Not accessible to Accountants
(10.51.1.0/24)</a></li>
        <li><a href="http://10.52.1.1/vendors">Accessible to Administrators
(10.52.1.0/24)</a></li>
        <li><a href="http://10.53.1.1/vendors">Accessible to Programmers (10.53.1.0/24) but not
*.gif or *.jpg files</a></li>
    </ul>

    <!-- Task 4 - Accountants -->
    <h3>Task 4 - Accountants website</h3>
    <ul>
        <li><a href="http://10.50.1.1/accountants">Not accessible to Vendors (10.50.1.0/24) but
not *.html files</a></li>
        <li><a href="http://10.51.1.1/accountants">Accessible to Accountants
(10.51.1.0/24)</a></li>
        <li><a href="http://10.52.1.1/accountants">Accessible to Administrators
(10.52.1.0/24)</a></li>
        <li><a href="http://10.53.1.1/accountants">Accessible to Programmers (10.53.1.0/24) but
not *.gif or *.jpg files</a></li>
    </ul>

    <!-- Task 4 - Programmers -->
    <h3>Task 4 - Programmers website</h3>
    <ul>
        <li><a href="http://10.50.1.1/programmers">Not accessible to Vendors
(10.50.1.0/24)</a></li>
        <li><a href="http://10.51.1.1/programmers">Not accessible to Accountants
(10.51.1.0/24)</a></li>
        <li><a href="http://10.52.1.1/programmers">Accessible to Administrators
(10.52.1.0/24)</a></li>
        <li><a href="http://10.53.1.1/programmers">Accessible to Programmers
(10.53.1.0/24)</a></li>
    </ul>

    <!-- Task 4 - Administrators -->
    <h3>Task 4 - Administrators website</h3>
    <ul>
        <li><a href="http://10.50.1.1/administrators">Not accessible to Vendors
(10.50.1.0/24)</a></li>
        <li><a href="http://10.51.1.1/administrators">Not accessible to Accountants
(10.51.1.0/24)</a></li>
        <li><a href="http://10.52.1.1/administrators">Accessible to Administrators
(10.52.1.0/24)</a></li>
        <li><a href="http://10.53.1.1/administrators">Accessible to Programmers (10.53.1.0/24)
but not *.gif or *.jpg files</a></li>
```

```
    </ul>

    <img src="example.jpg" alt="Example Image"><br>
</body>
</html>
```



6.3 Preparation

6.3.1 Create directories and files

6.3.1.1 Requirements

1. Creates Directories:

- /var/www/htdocs/vendors
- /var/www/htdocs/accountants
- /var/www/htdocs/administrators
- /var/www/htdocs/programmers

2. Creates Files:

- Each directory contains .html, .txt, and image files (Photo1.gif, Photo1.jpg).

- Files include sample content where required.

Directory	Files
/var/www/htdocs/vendors	vendors.html, doc.txt, Photo1.gif, Photo1.jpg
/var/www/htdocs/accountants	accountants.html, Photo1.gif, Photo1.jpg
/var/www/htdocs/administrators	administrators.html, Photo1.gif, Photo1.jpg
/var/www/htdocs/programmers	programmers.html, Photo1.gif, Photo1.jpg

3. Sets Permissions:

- Directories have 755 (read, write, and execute for owner; read and execute for group and others).
- Files have 644 (read/write for owner; read-only for group and others).

4. Adjusts Ownership:

- Directories and files are set to apache:apache for web server compatibility, as recommended for production environments.

6.3.1.2 Bash Script

A) Create the script in /home/mperez/Documents

cd /home/mperez/Documents

pwd

vim setup_web_directories.sh

```
#!/bin/bash
```

```
# Create directories
echo "Creating directories..."
mkdir -p /var/www/htdocs/{vendors,accountants,administrators,programmers}
echo "Directories created successfully!"
tree /var/www

# Create files for Vendors
echo "Creating files for Vendors..."
echo "<h1>Vendors Page</h1>" > /var/www/htdocs/vendors/vendors.html
echo "Vendor Document" > /var/www/htdocs/vendors/doc.txt
touch /var/www/htdocs/vendors/Photo1.gif /var/www/htdocs/vendors/Photo1.jpg
echo "Vendors files created successfully!"
tree /var/www/htdocs/vendors

# Create files for Accountants
echo "Creating files for Accountants..."
echo "<h1>Accountants Page</h1>" > /var/www/htdocs/accountants/accountants.html
touch /var/www/htdocs/accountants/Photo1.gif /var/www/htdocs/accountants/Photo1.jpg
echo "Accountants files created successfully!"
tree /var/www/htdocs/accountants

# Create files for Administrators
echo "Creating files for Administrators..."
echo "<h1>Administrators Page</h1>" > /var/www/htdocs/administrators/administrators.html
touch /var/www/htdocs/administrators/Photo1.gif
/var/www/htdocs/administrators/Photo1.jpg
echo "Administrators files created successfully!"
tree /var/www/htdocs/administrators

# Create files for Programmers
echo "Creating files for Programmers..."
echo "<h1>Programmers Page</h1>" > /var/www/htdocs/programmers/programmers.html
touch /var/www/htdocs/programmers/Photo1.gif /var/www/htdocs/programmers/Photo1.jpg
echo "Programmers files created successfully!"
tree /var/www/htdocs/programmers

# Set permissions
echo "Setting permissions..."
chown -R root:root /var/www/htdocs
chmod -R 755 /var/www/htdocs
chmod -R 644 /var/www/htdocs/*/*
echo "Permissions set successfully!"

# Adjust ownership for Apache (optional step for production environments)
echo "Adjusting ownership for Apache..."
chown -R apache:apache /var/www/htdocs
echo "Ownership adjusted successfully!"

# Confirm completion
echo "All tasks completed successfully!"
```

B) Make script executable:

chmod +x setup_web_directories.sh

C) Run the script as a superuser to ensure proper permissions and ownership:

sudo ./setup_web_directories.sh

```
[root@server1 Documents]# chmod +x setup_web_directories.sh
[root@server1 Documents]# sudo ./setup_web_directories.sh
Creating directories...
Directories created successfully!
/var/www
├── cgi-bin
├── htdocs
│   ├── accountants
│   ├── administrators
│   ├── programmers
│   └── vendors
├── html
└── html_project1
    ├── example.jpg
    ├── index.html
    ├── index.html_bkp
    ├── Project1
    │   ├── P1_image.gif
    │   ├── P1_logo.gif
    │   ├── P1_notes.txt
    │   ├── P1_secret.doc
    │   ├── P1_secret.html
    │   ├── P1_secret.txt
    │   ├── P1_test.html
    │   ├── project1.html
    │   ├── secret.doc
    │   ├── secret.txt
    │   └── test.html
    ├── Project2
```

```
    ├── Project2
    │   ├── P2_image.gif
    │   ├── P2_logo.gif
    │   ├── P2_notes.txt
    │   ├── P2_secret.doc
    │   ├── P2_secret.html
    │   ├── P2_secret.txt
    │   ├── P2_test.html
    │   ├── project2.html
    │   ├── secret.doc
    │   ├── secret.txt
    │   └── test.html
    ├── Project3
    │   ├── P3_image.gif
    │   ├── P3_logo.gif
    │   ├── P3_notes.txt
    │   ├── P3_secret.doc
    │   ├── P3_secret.html
    │   ├── P3_secret.txt
    │   ├── P3_test.html
    │   ├── project3.html
    │   ├── secret.doc
    │   ├── secret.txt
    │   └── test.html
    ├── Project4
```

```
    ├── Project4
    │   ├── P4_image.gif
    │   ├── P4_logo.gif
    │   ├── P4_notes.txt
    │   ├── P4_secret.doc
    │   ├── P4_secret.html
    │   ├── P4_secret.txt
    │   ├── P4_test.html
    │   ├── project4.html
    │   ├── secret.doc
    │   ├── secret.txt
    │   └── test.html
    ├── secure1
    │   └── index.html
    ├── secure2
    │   └── index.html
    ├── secure3
    │   └── index.html
    ├── secure4
    │   └── index.html
    ├── secure5
    │   └── index.html
    ├── secure6
    │   └── index.html
    └── secure7
        └── index.html
```

```
Vendors files created successfully!
/var/www/htdocs/vendors
├── doc.txt
├── Photo1.gif
├── Photo1.jpg
└── vendors.html

0 directories, 4 files
Creating files for Accountants...
Accountants files created successfully!
/var/www/htdocs/accountants
├── accountants.html
├── Photo1.gif
└── Photo1.jpg

0 directories, 3 files
Creating files for Administrators...
Administrators files created successfully!
/var/www/htdocs/administrators
├── administrators.html
├── Photo1.gif
└── Photo1.jpg

0 directories, 3 files
Creating files for Programmers...
Programmers files created successfully!
/var/www/htdocs/programmers
├── Photo1.gif
├── Photo1.jpg
└── programmers.html

0 directories, 3 files
Setting permissions...
Permissions set successfully!
Adjusting ownership for Apache...
Ownership adjusted successfully!
All tasks completed successfully!
[root@server1 Documents]#
```

D) List created files

tree /var/www/htdocs

```
[root@server1 Documents]#
[root@server1 Documents]# tree /var/www/htdocs/
/var/www/htdocs/
├── accountants
│   ├── accountants.html
│   ├── Photo1.gif
│   └── Photo1.jpg
├── administrators
│   ├── administrators.html
│   ├── Photo1.gif
│   └── Photo1.jpg
├── programmers
│   ├── Photo1.gif
│   ├── Photo1.jpg
│   └── programmers.html
└── vendors
    ├── doc.txt
    ├── Photo1.gif
    ├── Photo1.jpg
    └── vendors.html

4 directories, 13 files
[root@server1 Documents]#
```

ls -lqrtha -R /var/www/htdocs/

```
[root@server1 Documents]# ls -lqrtha -R /var/www/htdocs/
/var/www/htdocs/:
total 0
drwxr-xr-x. 2 apache apache 77 Apr 21 02:21 vendors
drwxr-xr-x. 2 apache apache 66 Apr 21 02:21 programmers
drwxr-xr-x. 2 apache apache 69 Apr 21 02:21 administrators
drwxr-xr-x. 2 apache apache 66 Apr 21 02:21 accountants
drwxr-xr-x. 6 root   root   68 Apr 21 02:21 ..
drwxr-xr-x. 6 apache apache 81 Apr 21 02:21 .

/var/www/htdocs/vendors:
total 8.0K
-rw-r--r--. 1 apache apache 22 Apr 21 02:21 vendors.html
-rw-r--r--. 1 apache apache  0 Apr 21 02:21 Photo1.jpg
-rw-r--r--. 1 apache apache  0 Apr 21 02:21 Photo1.gif
-rw-r--r--. 1 apache apache 16 Apr 21 02:21 doc.txt
drwxr-xr-x. 6 apache apache 81 Apr 21 02:21 ..
drwxr-xr-x. 2 apache apache 77 Apr 21 02:21 .

/var/www/htdocs/programmers:
total 4.0K
-rw-r--r--. 1 apache apache 26 Apr 21 02:21 programmers.html
-rw-r--r--. 1 apache apache  0 Apr 21 02:21 Photo1.jpg
-rw-r--r--. 1 apache apache  0 Apr 21 02:21 Photo1.gif
drwxr-xr-x. 6 apache apache 81 Apr 21 02:21 ..
drwxr-xr-x. 2 apache apache 66 Apr 21 02:21 .

/var/www/htdocs/administrators:
total 4.0K
-rw-r--r--. 1 apache apache  0 Apr 21 02:21 Photo1.jpg
-rw-r--r--. 1 apache apache  0 Apr 21 02:21 Photo1.gif
-rw-r--r--. 1 apache apache 29 Apr 21 02:21 administrators.html
drwxr-xr-x. 6 apache apache 81 Apr 21 02:21 ..
drwxr-xr-x. 2 apache apache 69 Apr 21 02:21 .

/var/www/htdocs/accountants:
total 4.0K
-rw-r--r--. 1 apache apache  0 Apr 21 02:21 Photo1.jpg
-rw-r--r--. 1 apache apache  0 Apr 21 02:21 Photo1.gif
-rw-r--r--. 1 apache apache 26 Apr 21 02:21 accountants.html
drwxr-xr-x. 6 apache apache 81 Apr 21 02:21 ..
drwxr-xr-x. 2 apache apache 66 Apr 21 02:21 .
[root@server1 Documents]#
```

6.3.2 Updated Configuration for /etc/httpd/conf/httpd.conf

1. Write directives according to the requirements

   Add these directives just before the </IfModule> closing tag for dir_module

   vim /etc/httpd/conf/httpd.conf

```
# Task 4 - Authorization Configuration

# Ensure parent directory access for aliases
<Directory "/var/www">
    Options None
    AllowOverride None
    Require all granted
</Directory>

# Allow access to Task 4 directories by default
<Directory "/var/www/htdocs">
    Options Indexes
    AllowOverride None
</Directory>
```

```
# Define aliases for Task 4 directories
Alias /vendors "/var/www/htdocs/vendors"
Alias /accountants "/var/www/htdocs/accountants"
Alias /administrators "/var/www/htdocs/administrators"
Alias /programmers "/var/www/htdocs/programmers"

# Vendors Directory
<Directory "/var/www/htdocs/vendors">
    Options +Indexes
    IndexOptions FancyIndexing
    AllowOverride None
    <RequireAny>
        Require ip 10.50.1.0/24
        Require ip 10.52.1.0/24
        Require ip 10.53.1.0/24
    </RequireAny>
    # Deny *.gif and *.jpg for Programmers in this directory
    <FilesMatch "\.(gif|jpg)$">
        <RequireAll>
            Require all granted
            Require not ip 10.53.1.0/24
        </RequireAll>
    </FilesMatch>
</Directory>

<Directory "/var/www/htdocs/accountants">
    Options +Indexes
    IndexOptions FancyIndexing
    AllowOverride None

    # Allow only Accountants, Administrators, and Programmers
    <RequireAny>
        Require ip 10.51.1.0/24
        Require ip 10.52.1.0/24
        Require ip 10.53.1.0/24
    </RequireAny>

    # Block Accountants from HTML files
    <FilesMatch "\.html$">
        <RequireAll>
            Require all granted
            Require not ip 10.51.1.0/24
        </RequireAll>
    </FilesMatch>

    # Block Programmers from image files
    <FilesMatch "\.(gif|jpg)$">
        <RequireAll>
            Require all granted
            Require not ip 10.53.1.0/24
        </RequireAll>
    </FilesMatch>

    DirectoryIndex disabled
</Directory>

# Administrators Directory
```

```
<Directory "/var/www/htdocs/administrators">
    Options +Indexes
    IndexOptions FancyIndexing
    AllowOverride None
    <RequireAny>
        Require ip 10.52.1.0/24
        Require ip 10.53.1.0/24
    </RequireAny>
    # Deny *.gif and *.jpg for Programmers in this directory
    <FilesMatch "\.(gif|jpg)$">
        <RequireAll>
            Require all granted
            Require not ip 10.53.1.0/24
        </RequireAll>
    </FilesMatch>
</Directory>

# Programmers Directory
<Directory "/var/www/htdocs/programmers">
    Options +Indexes
    IndexOptions FancyIndexing
    AllowOverride None
    <RequireAny>
        Require ip 10.52.1.0/24
        Require ip 10.53.1.0/24
    </RequireAny>
    # Deny *.gif and *.jpg for Programmers in this directory
    <FilesMatch "\.(gif|jpg)$">
        <RequireAll>
            Require all granted
            Require not ip 10.53.1.0/24
        </RequireAll>
    </FilesMatch>
</Directory>
```

```
# Global restriction for all .txt files across all directories
<Files "*.txt">
    Require all denied
</Files>

# Task 4 - Authorization Configuration

# Ensure parent directory access for aliases
<Directory "/var/www">
    Options None
    AllowOverride None
    Require all granted
</Directory>

# Allow access to Task 4 directories by default
<Directory "/var/www/htdocs">
    Options Indexes
    AllowOverride None
</Directory>

# Define aliases for Task 4 directories
Alias /vendors "/var/www/htdocs/vendors"
Alias /accountants "/var/www/htdocs/accountants"
Alias /administrators "/var/www/htdocs/administrators"
Alias /programmers "/var/www/htdocs/programmers"

# Vendors Directory
<Directory "/var/www/htdocs/vendors">
    Options +Indexes
    IndexOptions FancyIndexing
```

```
# Vendors Directory
<Directory "/var/www/htdocs/vendors">
    Options +Indexes
    IndexOptions FancyIndexing
    AllowOverride None
    <RequireAny>
        Require ip 10.50.1.0/24
        Require ip 10.52.1.0/24
        Require ip 10.53.1.0/24
    </RequireAny>
    # Deny *.gif and *.jpg for Programmers in this directory
    <FilesMatch "\.(gif|jpg)$">
        <RequireAll>
            Require all granted
            Require not ip 10.53.1.0/24
        </RequireAll>
    </FilesMatch>
</Directory>
```

```
<Directory "/var/www/htdocs/accountants">
    Options +Indexes
    IndexOptions FancyIndexing
    AllowOverride None

    # Allow only Accountants, Administrators, and Programmers
    <RequireAny>
        Require ip 10.51.1.0/24
        Require ip 10.52.1.0/24
        Require ip 10.53.1.0/24
    </RequireAny>

    # Block Accountants from HTML files
    <FilesMatch "\.html$">
        <RequireAll>
            Require all granted
            Require not ip 10.51.1.0/24
        </RequireAll>
    </FilesMatch>

    # Block Programmers from image files
    <FilesMatch "\.(gif|jpg)$">
        <RequireAll>
            Require all granted
            Require not ip 10.53.1.0/24
        </RequireAll>
    </FilesMatch>

    DirectoryIndex disabled
</Directory>
```

```
# Administrators Directory
<Directory "/var/www/htdocs/administrators">
    Options +Indexes
    IndexOptions FancyIndexing
    AllowOverride None
    <RequireAny>
        Require ip 10.52.1.0/24
        Require ip 10.53.1.0/24
    </RequireAny>
    # Deny *.gif and *.jpg for Programmers in this directory
    <FilesMatch "\.(gif|jpg)$">
        <RequireAll>
            Require all granted
            Require not ip 10.53.1.0/24
        </RequireAll>
    </FilesMatch>
</Directory>

# Programmers Directory
<Directory "/var/www/htdocs/programmers">
    Options +Indexes
    IndexOptions FancyIndexing
    AllowOverride None
    <RequireAny>
        Require ip 10.52.1.0/24
        Require ip 10.53.1.0/24
    </RequireAny>
    # Deny *.gif and *.jpg for Programmers in this directory
    <FilesMatch "\.(gif|jpg)$">
        <RequireAll>
            Require all granted
            Require not ip 10.53.1.0/24
        </RequireAll>
    </FilesMatch>
</Directory>

<IfModule dir_module>
    DirectoryIndex index.html
</IfModule>
```

2. Verify configuration

httpd -t

```
[root@server1 Documents]# httpd -t
Syntax OK
```

Expected syntax is ok , if any error go back and correct it.

3. Reload Apache:

systemctl reload httpd

```
[root@server1 ~]# systemctl reload httpd
[root@server1 ~]#
```

6.4 Testing

6.4.1 Manual test on AlmaLinux server

6.4.1.1 Manual test on Server with a script

1. Create an script on AlmaLinux to do manual testing according to the following table:

Test Case	Test Case	Command	Expected Result
TC001	Vendors access to /vendors	curl -s -o /dev/null -w 200 --interface 10.50.1.1 http://10.50.1.1/vendors/	200
TC002	Vendors access to vendors.html	curl -s -o /dev/null -w 200 --interface 10.50.1.1 http://10.50.1.1/vendors/vendors.html	200
TC003	Vendors access to Photo1.gif	curl -s -o /dev/null -w 200 --interface 10.50.1.1 http://10.50.1.1/vendors/Photo1.gif	200
TC004	Accountants access to /vendors	curl -s -o /dev/null -w 403 --interface 10.51.1.1 http://10.50.1.1/vendors/	403
TC005	Administrators access to /vendors	curl -s -o /dev/null -w 200 --interface 10.52.1.1 http://10.50.1.1/vendors/	200
TC006	Administrators access to vendors.html	curl -s -o /dev/null -w 200 --interface 10.52.1.1 http://10.50.1.1/vendors/vendors.html	200
TC007	Administrators access to Photo1.gif	curl -s -o /dev/null -w 200 --interface 10.52.1.1 http://10.50.1.1/vendors/Photo1.gif	200
TC008	Programmers access to /vendors	curl -s -o /dev/null -w 200 --interface 10.53.1.1 http://10.50.1.1/vendors/	200
TC009	Programmers access to vendors.html	curl -s -o /dev/null -w 200 --interface 10.53.1.1 http://10.50.1.1/vendors/vendors.html	200
TC010	Programmers access to Photo1.gif	curl -s -o /dev/null -w 403 --interface 10.53.1.1 http://10.50.1.1/vendors/Photo1.gif	403
TC011	Vendors access to /accountants	curl -s -o /dev/null -w 403 --interface 10.50.1.1 http://10.50.1.1/accountants/	403
TC012	Accountants access to /accountants	curl -s -o /dev/null -w 200 --interface 10.51.1.1 http://10.50.1.1/accountants/	200
TC013	Accountants access to accountants.html	curl -s -o /dev/null -w 403 --interface 10.51.1.1 http://10.50.1.1/accountants/accountants.html	403
TC014	Accountants access to Photo1.gif	curl -s -o /dev/null -w 200 --interface 10.51.1.1 http://10.50.1.1/accountants/Photo1.gif	200
TC015	Administrators access to /accountants	curl -s -o /dev/null -w 200 --interface 10.52.1.1 http://10.50.1.1/accountants/	200
TC016	Administrators access to accountants.html	curl -s -o /dev/null -w 200 --interface 10.52.1.1 http://10.50.1.1/accountants/accountants.html	200
TC017	Administrators access to Photo1.gif	curl -s -o /dev/null -w 200 --interface 10.52.1.1 http://10.50.1.1/accountants/Photo1.gif	200
TC018	Programmers access to /accountants	curl -s -o /dev/null -w 200 --interface 10.53.1.1 http://10.50.1.1/accountants/	200
TC019	Programmers access to accountants.html	curl -s -o /dev/null -w 200 --interface 10.53.1.1 http://10.50.1.1/accountants/accountants.html	200
TC020	Programmers access to Photo1.gif	curl -s -o /dev/null -w 403 --interface 10.53.1.1 http://10.50.1.1/accountants/Photo1.gif	403
TC021	Vendors access to /programmers	curl -s -o /dev/null -w 403 --interface 10.50.1.1 http://10.50.1.1/programmers/	403
TC022	Accountants access to /programmers	curl -s -o /dev/null -w 403 --interface 10.51.1.1 http://10.50.1.1/programmers/	403
TC023	Administrators access to /programmers	curl -s -o /dev/null -w 200 --interface 10.52.1.1 http://10.50.1.1/programmers/	200
TC024	Administrators access to programmers.html	curl -s -o /dev/null -w 200 --interface 10.52.1.1 http://10.50.1.1/programmers/programmers.html	200
TC025	Administrators access to Photo1.gif	curl -s -o /dev/null -w 200 --interface 10.52.1.1 http://10.50.1.1/programmers/Photo1.gif	200
TC026	Programmers access to /programmers	curl -s -o /dev/null -w 200 --interface 10.53.1.1 http://10.50.1.1/programmers/	200
TC027	Programmers access to programmers.html	curl -s -o /dev/null -w 200 --interface 10.53.1.1 http://10.50.1.1/programmers/programmers.html	200
TC028	Programmers access to Photo1.gif	curl -s -o /dev/null -w 403 --interface 10.53.1.1 http://10.50.1.1/programmers/Photo1.gif	403
TC029	Vendors access to /administrators	curl -s -o /dev/null -w 403 --interface 10.50.1.1 http://10.50.1.1/administrators/	403
TC030	Accountants access to /administrators	curl -s -o /dev/null -w 403 --interface 10.51.1.1 http://10.50.1.1/administrators/	403
TC031	Administrators access to /administrators	curl -s -o /dev/null -w 200 --interface 10.52.1.1 http://10.50.1.1/administrators/	200
TC032	Administrators access to administrators.html	curl -s -o /dev/null -w 200 --interface 10.52.1.1 http://10.50.1.1/administrators/administrators.html	200
TC033	Administrators access to Photo1.gif	curl -s -o /dev/null -w 200 --interface 10.52.1.1 http://10.50.1.1/administrators/Photo1.gif	200
TC034	Programmers access to /administrators	curl -s -o /dev/null -w 200 --interface 10.53.1.1 http://10.50.1.1/administrators/	200
TC035	Programmers access to administrators.html	curl -s -o /dev/null -w 200 --interface 10.53.1.1 http://10.50.1.1/administrators/administrators.html	200
TC036	Programmers access to Photo1.gif	curl -s -o /dev/null -w 403 --interface 10.53.1.1 http://10.50.1.1/administrators/Photo1.gif	403

The script tests access to four directories (/vendors, /accountants, /programmers, /administrators) on a web server (10.50.1.1) for four user groups: Vendors, Accountants, Administrators, and Programmers. Each group is represented by a client IP:

- Vendors: 10.50.1.1
- Accountants: 10.51.1.1
- Administrators: 10.52.1.1
- Programmers: 10.53.1.1

The script sends HTTP requests using curl, checks the HTTP status code (e.g., 200 for allowed, 403 for denied), and logs the results to a file (/root/task4_test_results.log). The modifications add a test case number (e.g., TC001) to each test and a summary table at the end showing the TC number and PASS/FAIL verdict.

The curl command used in the script sends an HTTP request to a server while specifying certain options to control its behavior. Here's a breakdown of what each part of the command does:

curl -s -o /dev/null -w "%{http_code}" --interface 10.50.1.1 http://10.50.1.1/vendors/

Explanation

1. **curl**: A command-line tool to transfer data to or from a server using various protocols (e.g., HTTP, HTTPS, FTP).
2. **-s (Silent mode)**: Suppresses the progress bar and error messages, making the command output cleaner.
3. **-o /dev/null**: Redirects the output of the curl request to /dev/null, effectively discarding it. This is useful when you only want the status or metadata.
4. **-w "%{http_code}"**: Specifies what information to output. Here, it prints the HTTP status code (e.g., 200 for success, 403 for forbidden).
5. **--interface 10.50.1.1**: Uses the network interface with the IP address 10.50.1.1 for the HTTP request.
6. **http://10.50.1.1/vendors/**: The URL to which the HTTP request is sent.

Purpose

This command checks the HTTP response code for the URL http://10.50.1.1/vendors/ using the network interface 10.50.1.1. It is useful for testing server responses and ensuring the correct status codes are returned based on access permissions.

For example:

- A 200 code indicates success.
- A 403 code indicates forbidden access.

If same command is to be done manually , remove the part for automation and log

curl ~~-s -o /dev/null -w 200~~ --interface 10.50.1.1 http://10.50.1.1/vendors/

Run command manually and put option -v to see the output

curl -v --interface 10.50.1.1 http://10.50.1.1/vendors/

```
[mperez@server1 ~]$ curl -v  --interface 10.50.1.1 http://10.50.1.1/vendors/
*   Trying 10.50.1.1:80...
* Name '10.50.1.1' family 2 resolved to '10.50.1.1' family 2
* Local port: 0
* Connected to 10.50.1.1 (10.50.1.1) port 80 (#0)
> GET /vendors/ HTTP/1.1
> Host: 10.50.1.1
> User-Agent: curl/7.76.1
> Accept: */*
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 200 OK
< Date: Tue, 22 Apr 2025 01:34:16 GMT
< Server: Apache/2.4.62 (AlmaLinux)
< Content-Length: 657
< Content-Type: text/html;charset=ISO-8859-1
<
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
<html>
 <head>
  <title>Index of /vendors</title>
 </head>
 <body>
<h1>Index of /vendors</h1>
<pre>      <a href="?C=N;O=D">Name</a>                    <a href="?C=M;O=A">Last modified</a>      <a href="?C=S;O=A">Size</a>  <a href="?C=D;O=A">Description</a><hr>
    <a href="/">Parent Directory</a>                             -
     <a href="Photo1.gif">Photo1.gif</a>            2025-04-21 11:18    0
     <a href="Photo1.jpg">Photo1.jpg</a>            2025-04-21 11:18    0
     <a href="vendors.html">vendors.html</a>          2025-04-21 11:18   22
<hr></pre>
</body></html>
* Connection #0 to host 10.50.1.1 left intact
[mperez@server1 ~]$
```

3. Write script

cd /home/mperez/Documents

pwd

vim test_task4.sh

```
[root@server1 httpd]#
[root@server1 httpd]# cd /home/mperez/Documents
[root@server1 Documents]# pwd
/home/mperez/Documents
[root@server1 Documents]# vim test_task4.sh
```

```
#!/bin/bash

# Test script for Task 4 - Authorization
# Server IP
SERVER="10.50.1.1"

# Client IPs
VENDORS_IP="10.50.1.1"
ACCOUNTANTS_IP="10.51.1.1"
ADMINISTRATORS_IP="10.52.1.1"
PROGRAMMERS_IP="10.53.1.1"

# Log file
LOG_FILE="/root/task4_test_results.log"
echo "Task 4 Test Results - $(date)" > "$LOG_FILE"
echo "----------------------------------------" >> "$LOG_FILE"

# Function to test access
test_access() {
    local url="$1"
    local interface="$2"
    local expected_status="$3"
    local test_name="$4"

    # Get HTTP status code
    status=$(curl -s -o /dev/null -w "%{http_code}" --interface "$interface" "$url")
```

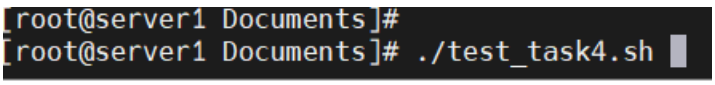
```
    # Log result
    if [ "$status" -eq "$expected_status" ]; then
        result="PASS"
    else
        result="FAIL"
    fi

    echo "Test: $test_name" >> "$LOG_FILE"
    echo "URL: $url" >> "$LOG_FILE"
    echo "Client IP: $interface" >> "$LOG_FILE"
    echo "Expected Status: $expected_status, Actual Status: $status" >> "$LOG_FILE"
    echo "Result: $result" >> "$LOG_FILE"
    echo "----------------------------------------" >> "$LOG_FILE"
}

# Task 4 - Vendors website
echo "Testing Vendors website..." | tee -a "$LOG_FILE"
test_access "http://$SERVER/vendors/" "$VENDORS_IP" 200 "Vendors access to /vendors"
test_access "http://$SERVER/vendors/vendors.html" "$VENDORS_IP" 200 "Vendors access to
vendors.html"
test_access "http://$SERVER/vendors/Photo1.gif" "$VENDORS_IP" 200 "Vendors access to
Photo1.gif"
test_access "http://$SERVER/vendors/" "$ACCOUNTANTS_IP" 403 "Accountants access to
/vendors"
test_access "http://$SERVER/vendors/" "$ADMINISTRATORS_IP" 200 "Administrators access to
/vendors"
test_access "http://$SERVER/vendors/vendors.html" "$ADMINISTRATORS_IP" 200
"Administrators access to vendors.html"
test_access "http://$SERVER/vendors/Photo1.gif" "$ADMINISTRATORS_IP" 200 "Administrators
access to Photo1.gif"
test_access "http://$SERVER/vendors/" "$PROGRAMMERS_IP" 200 "Programmers access to
/vendors"
test_access "http://$SERVER/vendors/vendors.html" "$PROGRAMMERS_IP" 200 "Programmers
access to vendors.html"
test_access "http://$SERVER/vendors/Photo1.gif" "$PROGRAMMERS_IP" 403 "Programmers
access to Photo1.gif"

# Task 4 - Accountants website
echo "Testing Accountants website..." | tee -a "$LOG_FILE"
test_access "http://$SERVER/accountants/" "$VENDORS_IP" 403 "Vendors access to
/accountants"
test_access "http://$SERVER/accountants/" "$ACCOUNTANTS_IP" 200 "Accountants access to
/accountants"
test_access "http://$SERVER/accountants/accountants.html" "$ACCOUNTANTS_IP" 403
"Accountants access to accountants.html"
test_access "http://$SERVER/accountants/Photo1.gif" "$ACCOUNTANTS_IP" 200 "Accountants
access to Photo1.gif"
test_access "http://$SERVER/accountants/" "$ADMINISTRATORS_IP" 200 "Administrators
access to /accountants"
test_access "http://$SERVER/accountants/accountants.html" "$ADMINISTRATORS_IP" 200
"Administrators access to accountants.html"
test_access "http://$SERVER/accountants/Photo1.gif" "$ADMINISTRATORS_IP" 200
"Administrators access to Photo1.gif"
test_access "http://$SERVER/accountants/" "$PROGRAMMERS_IP" 200 "Programmers access to
/accountants"
test_access "http://$SERVER/accountants/accountants.html" "$PROGRAMMERS_IP" 200
"Programmers access to accountants.html"
test_access "http://$SERVER/accountants/Photo1.gif" "$PROGRAMMERS_IP" 403 "Programmers
access to Photo1.gif"

# Task 4 - Programmers website
echo "Testing Programmers website..." | tee -a "$LOG_FILE"
```

```
test_access "http://$SERVER/programmers/" "$VENDORS_IP" 403 "Vendors access to
/programmers"
test_access "http://$SERVER/programmers/" "$ACCOUNTANTS_IP" 403 "Accountants access to
/programmers"
test_access "http://$SERVER/programmers/" "$ADMINISTRATORS_IP" 200 "Administrators
access to /programmers"
test_access "http://$SERVER/programmers/programmers.html" "$ADMINISTRATORS_IP" 200
"Administrators access to programmers.html"
test_access "http://$SERVER/programmers/Photo1.gif" "$ADMINISTRATORS_IP" 200
"Administrators access to Photo1.gif"
test_access "http://$SERVER/programmers/" "$PROGRAMMERS_IP" 200 "Programmers access to
/programmers"
test_access "http://$SERVER/programmers/programmers.html" "$PROGRAMMERS_IP" 200
"Programmers access to programmers.html"
test_access "http://$SERVER/programmers/Photo1.gif" "$PROGRAMMERS_IP" 403 "Programmers
access to Photo1.gif"

# Task 4 - Administrators website
echo "Testing Administrators website..." | tee -a "$LOG_FILE"
test_access "http://$SERVER/administrators/" "$VENDORS_IP" 403 "Vendors access to
/administrators"
test_access "http://$SERVER/administrators/" "$ACCOUNTANTS_IP" 403 "Accountants access
to /administrators"
test_access "http://$SERVER/administrators/" "$ADMINISTRATORS_IP" 200 "Administrators
access to /administrators"
test_access "http://$SERVER/administrators/administrators.html" "$ADMINISTRATORS_IP" 200
"Administrators access to administrators.html"
test_access "http://$SERVER/administrators/Photo1.gif" "$ADMINISTRATORS_IP" 200
"Administrators access to Photo1.gif"
test_access "http://$SERVER/administrators/" "$PROGRAMMERS_IP" 200 "Programmers access
to /administrators"
test_access "http://$SERVER/administrators/administrators.html" "$PROGRAMMERS_IP" 200
"Programmers access to administrators.html"
test_access "http://$SERVER/administrators/Photo1.gif" "$PROGRAMMERS_IP" 403
"Programmers access to Photo1.gif"

echo "Testing complete. Results logged to $LOG_FILE"
cat "$LOG_FILE"
```

4. Give execute permissions
   chmod +x test_task4.sh

5. Run the script

   ./test_task4



   At the end there is a table with a TC number and the indication of PASS or FAIL
   If test case FAIL go back and check.
   Repeat configuration changes until all test cases PASS.

   Example of successful run :

```
[root@server1 Documents]# ./test_task4.sh
Testing Vendors website...
```

```
Testing Accountants website...
Testing Programmers website...
Testing Administrators website...
Summary of Test Results
Testing complete. Results logged to /root/task4_test_results.log
Task 4 Test Results - Mon 21 Apr 2025 07:22:02 PM EDT
---------------------------------------
Testing Vendors website...
Test Case: TC001
Test: Vendors access to /vendors
URL: http://10.50.1.1/vendors/
Client IP: 10.50.1.1
Expected Status: 200, Actual Status: 200
Result: PASS
---------------------------------------
Test Case: TC002
Test: Vendors access to vendors.html
URL: http://10.50.1.1/vendors/vendors.html
Client IP: 10.50.1.1
Expected Status: 200, Actual Status: 200
Result: PASS
---------------------------------------
Test Case: TC003
Test: Vendors access to Photo1.gif
URL: http://10.50.1.1/vendors/Photo1.gif
Client IP: 10.50.1.1
Expected Status: 200, Actual Status: 200
Result: PASS
---------------------------------------
Test Case: TC004
Test: Accountants access to /vendors
URL: http://10.50.1.1/vendors/
Client IP: 10.51.1.1
Expected Status: 403, Actual Status: 403
Result: PASS
---------------------------------------
Test Case: TC005
Test: Administrators access to /vendors
URL: http://10.50.1.1/vendors/
Client IP: 10.52.1.1
Expected Status: 200, Actual Status: 200
Result: PASS
---------------------------------------
Test Case: TC006
Test: Administrators access to vendors.html
URL: http://10.50.1.1/vendors/vendors.html
Client IP: 10.52.1.1
Expected Status: 200, Actual Status: 200
Result: PASS
---------------------------------------
Test Case: TC007
Test: Administrators access to Photo1.gif
URL: http://10.50.1.1/vendors/Photo1.gif
Client IP: 10.52.1.1
Expected Status: 200, Actual Status: 200
Result: PASS
---------------------------------------
Test Case: TC008
Test: Programmers access to /vendors
URL: http://10.50.1.1/vendors/
Client IP: 10.53.1.1
Expected Status: 200, Actual Status: 200
Result: PASS
---------------------------------------
Test Case: TC009
Test: Programmers access to vendors.html
URL: http://10.50.1.1/vendors/vendors.html
Client IP: 10.53.1.1
Expected Status: 200, Actual Status: 200
Result: PASS
---------------------------------------
Test Case: TC010
Test: Programmers access to Photo1.gif
URL: http://10.50.1.1/vendors/Photo1.gif
Client IP: 10.53.1.1
Expected Status: 403, Actual Status: 403
Result: PASS
---------------------------------------
Testing Accountants website...
```

```
Test Case: TC011
Test: Vendors access to /accountants
URL: http://10.50.1.1/accountants/
Client IP: 10.50.1.1
Expected Status: 403, Actual Status: 403
Result: PASS
----------------------------------------
Test Case: TC012
Test: Accountants access to /accountants
URL: http://10.50.1.1/accountants/
Client IP: 10.51.1.1
Expected Status: 200, Actual Status: 200
Result: PASS
----------------------------------------
Test Case: TC013
Test: Accountants access to accountants.html
URL: http://10.50.1.1/accountants/accountants.html
Client IP: 10.51.1.1
Expected Status: 403, Actual Status: 403
Result: PASS
----------------------------------------
Test Case: TC014
Test: Accountants access to Photo1.gif
URL: http://10.50.1.1/accountants/Photo1.gif
Client IP: 10.51.1.1
Expected Status: 200, Actual Status: 200
Result: PASS
----------------------------------------
Test Case: TC015
Test: Administrators access to /accountants
URL: http://10.50.1.1/accountants/
Client IP: 10.52.1.1
Expected Status: 200, Actual Status: 200
Result: PASS
----------------------------------------
Test Case: TC016
Test: Administrators access to accountants.html
URL: http://10.50.1.1/accountants/accountants.html
Client IP: 10.52.1.1
Expected Status: 200, Actual Status: 200
Result: PASS
----------------------------------------
Test Case: TC017
Test: Administrators access to Photo1.gif
URL: http://10.50.1.1/accountants/Photo1.gif
Client IP: 10.52.1.1
Expected Status: 200, Actual Status: 200
Result: PASS
----------------------------------------
Test Case: TC018
Test: Programmers access to /accountants
URL: http://10.50.1.1/accountants/
Client IP: 10.53.1.1
Expected Status: 200, Actual Status: 200
Result: PASS
----------------------------------------
Test Case: TC019
Test: Programmers access to accountants.html
URL: http://10.50.1.1/accountants/accountants.html
Client IP: 10.53.1.1
Expected Status: 200, Actual Status: 200
Result: PASS
----------------------------------------
Test Case: TC020
Test: Programmers access to Photo1.gif
URL: http://10.50.1.1/accountants/Photo1.gif
Client IP: 10.53.1.1
Expected Status: 403, Actual Status: 403
Result: PASS
----------------------------------------
Testing Programmers website...
Test Case: TC021
Test: Vendors access to /programmers
URL: http://10.50.1.1/programmers/
Client IP: 10.50.1.1
Expected Status: 403, Actual Status: 403
Result: PASS
----------------------------------------
Test Case: TC022
```

```
Test: Accountants access to /programmers
URL: http://10.50.1.1/programmers/
Client IP: 10.51.1.1
Expected Status: 403, Actual Status: 403
Result: PASS
----------------------------------------
Test Case: TC023
Test: Administrators access to /programmers
URL: http://10.50.1.1/programmers/
Client IP: 10.52.1.1
Expected Status: 200, Actual Status: 200
Result: PASS
----------------------------------------
Test Case: TC024
Test: Administrators access to programmers.html
URL: http://10.50.1.1/programmers/programmers.html
Client IP: 10.52.1.1
Expected Status: 200, Actual Status: 200
Result: PASS
----------------------------------------
Test Case: TC025
Test: Administrators access to Photo1.gif
URL: http://10.50.1.1/programmers/Photo1.gif
Client IP: 10.52.1.1
Expected Status: 200, Actual Status: 200
Result: PASS
----------------------------------------
Test Case: TC026
Test: Programmers access to /programmers
URL: http://10.50.1.1/programmers/
Client IP: 10.53.1.1
Expected Status: 200, Actual Status: 200
Result: PASS
----------------------------------------
Test Case: TC027
Test: Programmers access to programmers.html
URL: http://10.50.1.1/programmers/programmers.html
Client IP: 10.53.1.1
Expected Status: 200, Actual Status: 200
Result: PASS
----------------------------------------
Test Case: TC028
Test: Programmers access to Photo1.gif
URL: http://10.50.1.1/programmers/Photo1.gif
Client IP: 10.53.1.1
Expected Status: 403, Actual Status: 403
Result: PASS
----------------------------------------
Testing Administrators website...
Test Case: TC029
Test: Vendors access to /administrators
URL: http://10.50.1.1/administrators/
Client IP: 10.50.1.1
Expected Status: 403, Actual Status: 403
Result: PASS
----------------------------------------
Test Case: TC030
Test: Accountants access to /administrators
URL: http://10.50.1.1/administrators/
Client IP: 10.51.1.1
Expected Status: 403, Actual Status: 403
Result: PASS
----------------------------------------
Test Case: TC031
Test: Administrators access to /administrators
URL: http://10.50.1.1/administrators/
Client IP: 10.52.1.1
Expected Status: 200, Actual Status: 200
Result: PASS
----------------------------------------
Test Case: TC032
Test: Administrators access to administrators.html
URL: http://10.50.1.1/administrators/administrators.html
Client IP: 10.52.1.1
Expected Status: 200, Actual Status: 200
Result: PASS
----------------------------------------
Test Case: TC033
Test: Administrators access to Photo1.gif
```

```
URL: http://10.50.1.1/administrators/Photo1.gif
Client IP: 10.52.1.1
Expected Status: 200, Actual Status: 200
Result: PASS
----------------------------------------
Test Case: TC034
Test: Programmers access to /administrators
URL: http://10.50.1.1/administrators/
Client IP: 10.53.1.1
Expected Status: 200, Actual Status: 200
Result: PASS
----------------------------------------
Test Case: TC035
Test: Programmers access to administrators.html
URL: http://10.50.1.1/administrators/administrators.html
Client IP: 10.53.1.1
Expected Status: 200, Actual Status: 200
Result: PASS
----------------------------------------
Test Case: TC036
Test: Programmers access to Photo1.gif
URL: http://10.50.1.1/administrators/Photo1.gif
Client IP: 10.53.1.1
Expected Status: 403, Actual Status: 403
Result: PASS
----------------------------------------
Summary of Test Results
----------------------------------------
Test Case | Result
----------|--------
TC001      | PASS
TC002      | PASS
TC003      | PASS
TC004      | PASS
TC005      | PASS
TC006      | PASS
TC007      | PASS
TC008      | PASS
TC009      | PASS
TC010      | PASS
TC011      | PASS
TC012      | PASS
TC013      | PASS
TC014      | PASS
TC015      | PASS
TC016      | PASS
TC017      | PASS
TC018      | PASS
TC019      | PASS
TC020      | PASS
TC021      | PASS
TC022      | PASS
TC023      | PASS
TC024      | PASS
TC025      | PASS
TC026      | PASS
TC027      | PASS
TC028      | PASS
TC029      | PASS
TC030      | PASS
TC031      | PASS
TC032      | PASS
TC033      | PASS
TC034      | PASS
TC035      | PASS
TC036      | PASS
----------------------------------------
[root@server1 Documents]#
```

6.4.2 Web browser test cases on Ubuntu client

Menu for Task 4

Task 4

Task 4 - Vendors website

- Accessible to Vendors (10.50.1.0/24))
- Not accessible to Accountants (10.51.1.0/24)
- Accessible to Administrators (10.52.1.0/24)
- Accessible to Programmers (10.53.1.0/24) but not *.gif or *.jpg files

Task 4 - Accountants website

- Not accessible to Vendors (10.50.1.0/24) but not *.html files
- Accessible to Accountants (10.51.1.0/24)
- Accessible to Administrators (10.52.1.0/24)
- Accessible to Programmers (10.53.1.0/24) but not *.gif or *.jpg files

Task 4 - Programmers website

- Not accessible to Vendors (10.50.1.0/24)
- Not accessible to Accountants (10.51.1.0/24)
- Accessible to Administrators (10.52.1.0/24)
- Accessible to Programmers (10.53.1.0/24)

Task 4 - Administrators website

- Not accessible to Vendors (10.50.1.0/24)
- Not accessible to Accountants (10.51.1.0/24)
- Accessible to Administrators (10.52.1.0/24)
- Accessible to Programmers (10.53.1.0/24) but not *.gif or *.jpg files

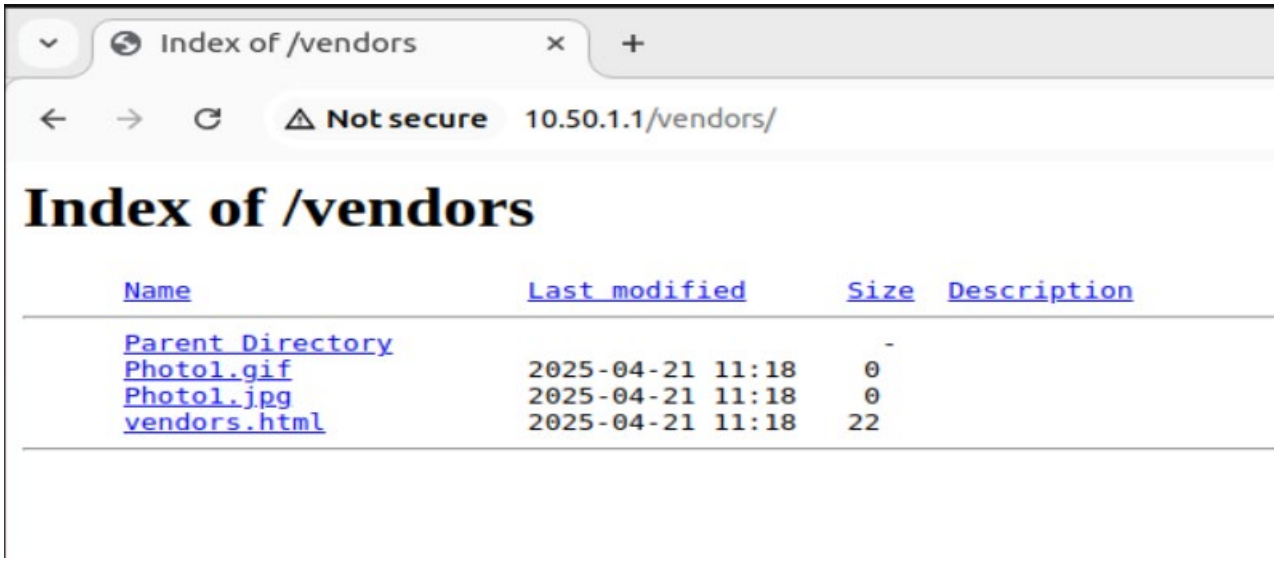
6.4.2.1 Task 4 – Vendors website testing

Task 4 - Vendors website

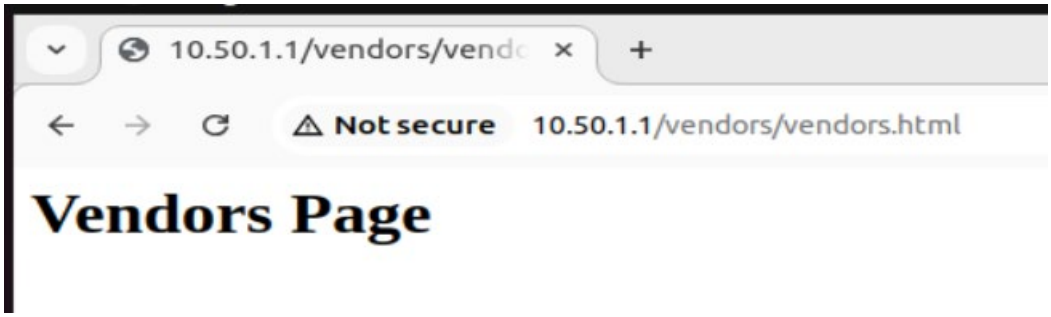
- Accessible to Vendors (10.50.1.0/24))
- Not accessible to Accountants (10.51.1.0/24)
- Accessible to Administrators (10.52.1.0/24)
- Accessible to Programmers (10.53.1.0/24) but not *.gif or *.jpg files

6.4.2.1.1 - Accessible to Vendors (10.50.1.0/24)

Select first choice in the menu and double click to get the next screen

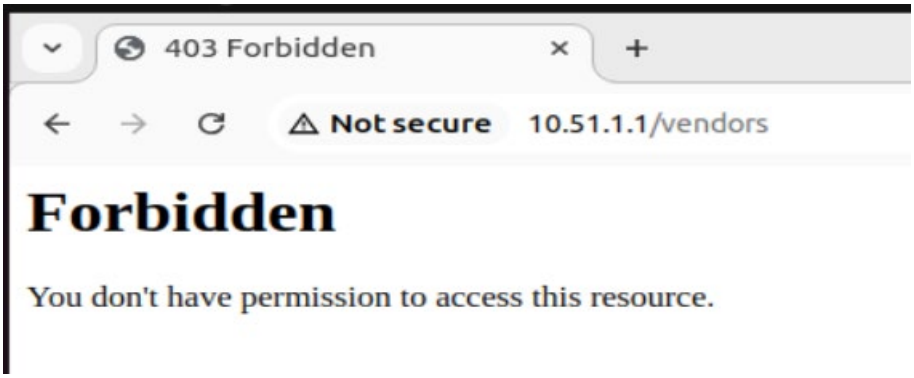


If you select **vendors.html** the following screen is displayed



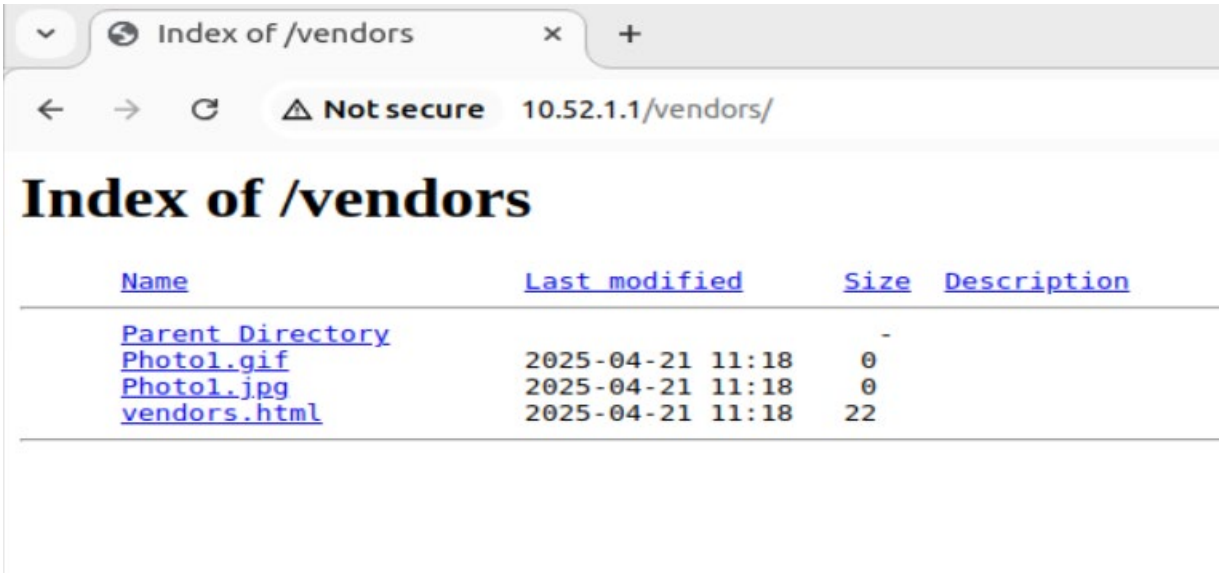
6.4.2.1.2 - Not accessible to Accountants (10.51.1.0/24)

Select second choice in the menu and double click to get the next screen.

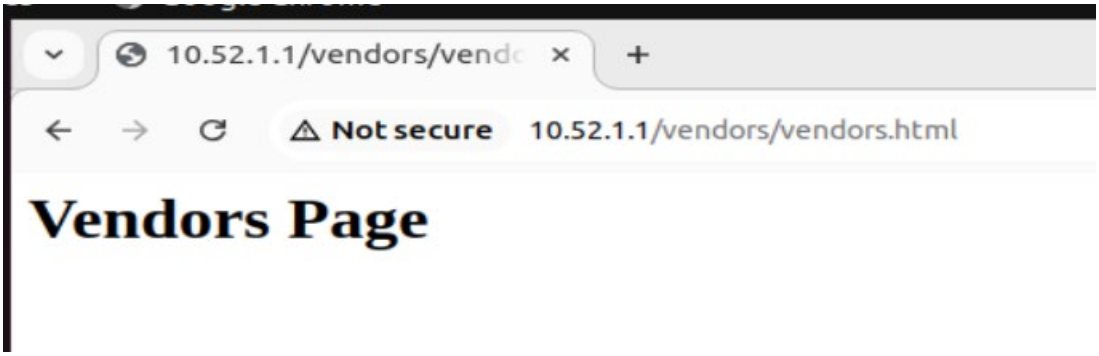


6.4.2.1.3 - Accessible to Administrators (10.52.1.0/24)

Select third choice in the menu and double click to get the next screen.

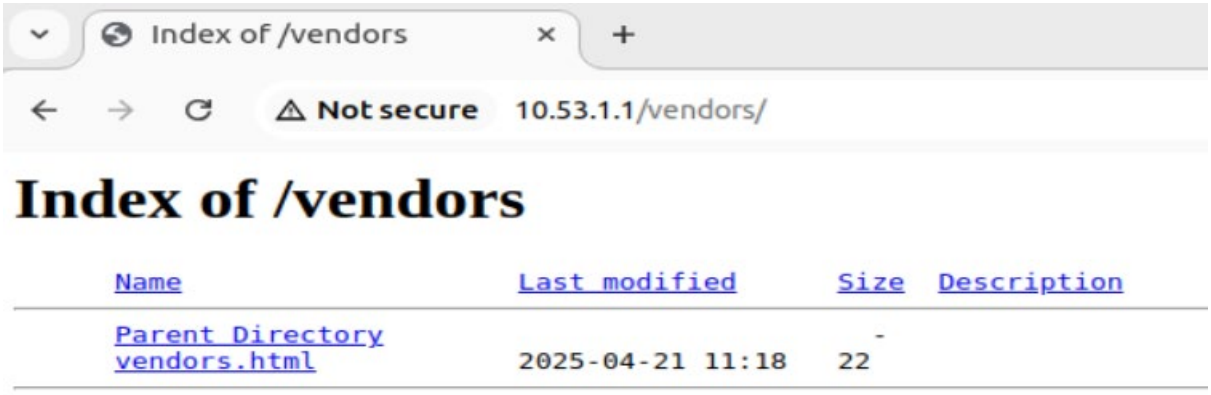


If you select **vendors.html** the following screen is displayed



6.4.2.1.4 - Accessible to Programmers (10.53.1.0/24) but not *.gif or *.jpg files

Select fourth choice in the menu and double click to get the next screen.



If you select **vendors.html** the following screen is displayed

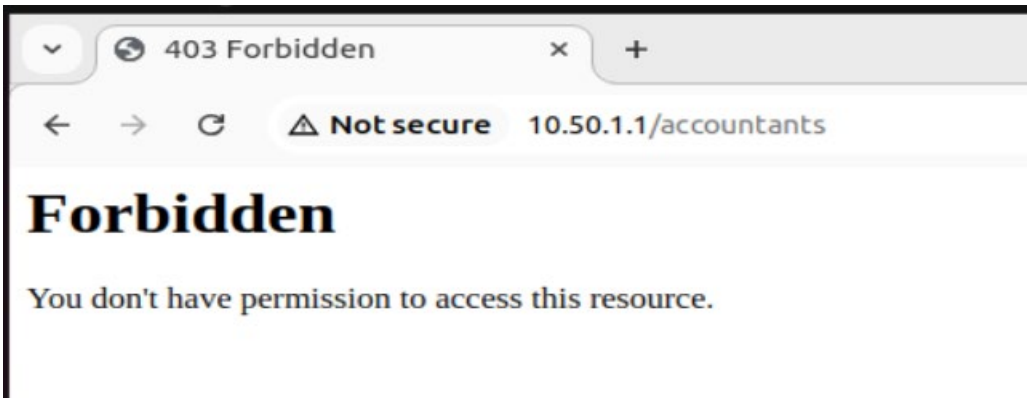
6.4.2.2 Task 4 – Accountants website testing

Task 4 - Accountants website

- Not accessible to Vendors (10.50.1.0/24) but not *.html files
- Accessible to Accountants (10.51.1.0/24)
- Accessible to Administrators (10.52.1.0/24)
- Accessible to Programmers (10.53.1.0/24) but not *.gif or *.jpg files

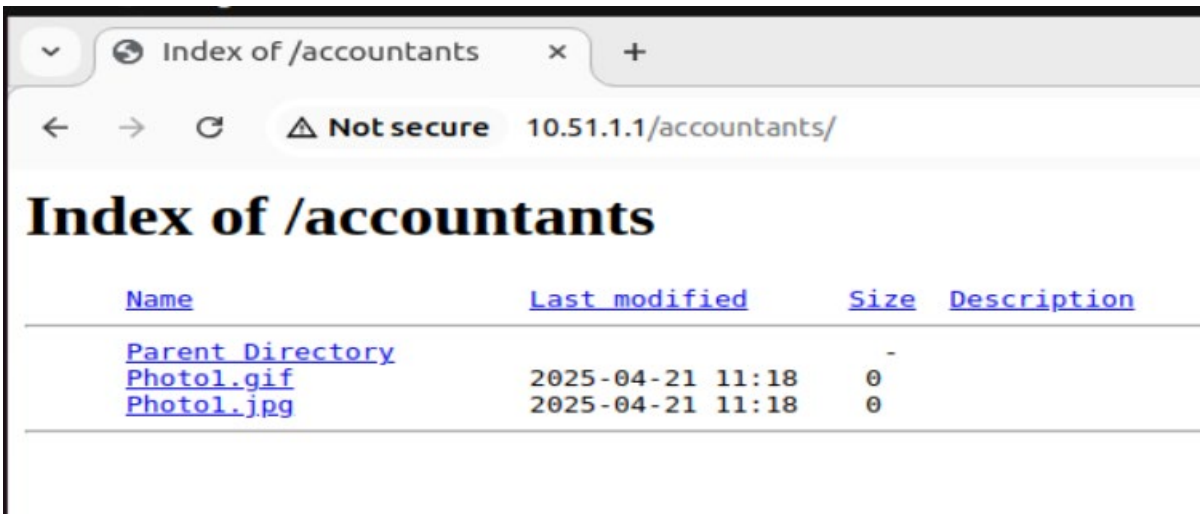
6.4.2.2.1 - Not accessible to Vendors (10.50.1.0/24) but not *.html files

Select first choice in the menu and double click to get the next screen.



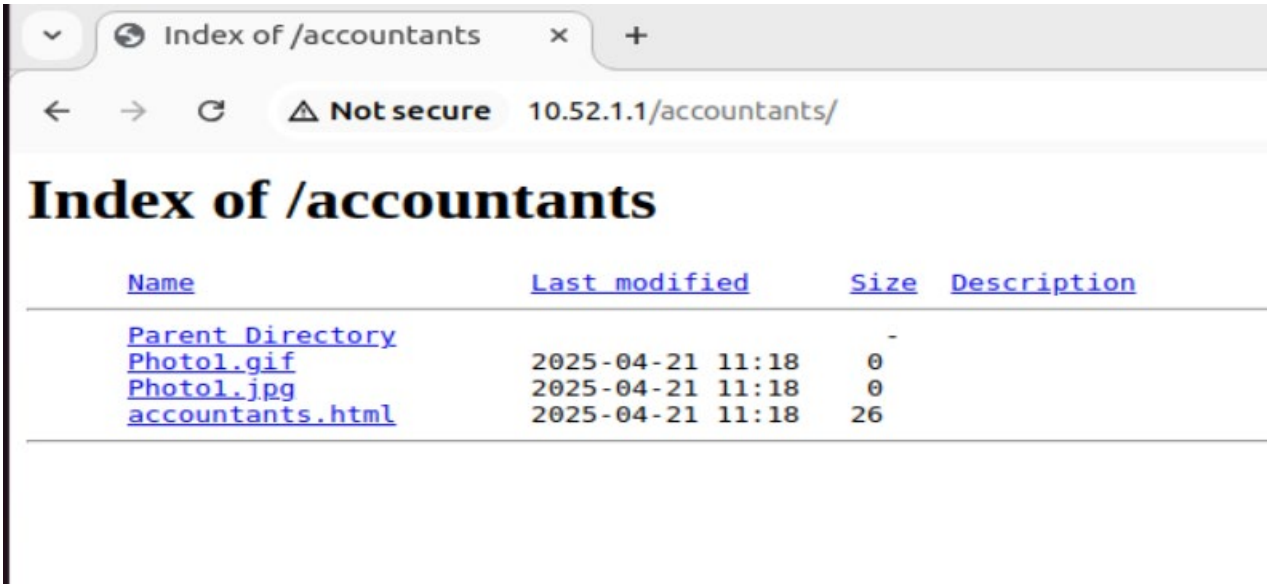
6.4.2.2.2 - Accessible to Accountants (10.51.1.0/24)

Select second choice in the menu and double click to get the next screen.

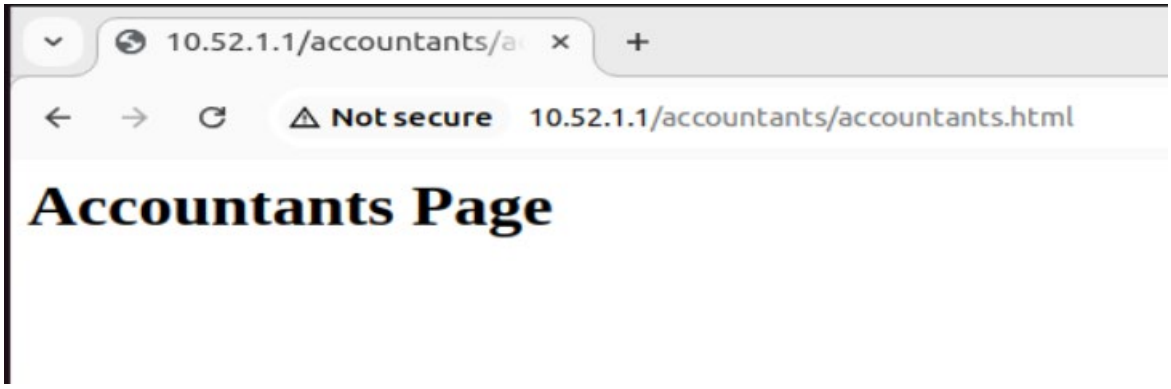


6.4.2.2.3 - Accessible to Administrators (10.52.1.0/24)

Select third choice in the menu and double click to get the next screen.

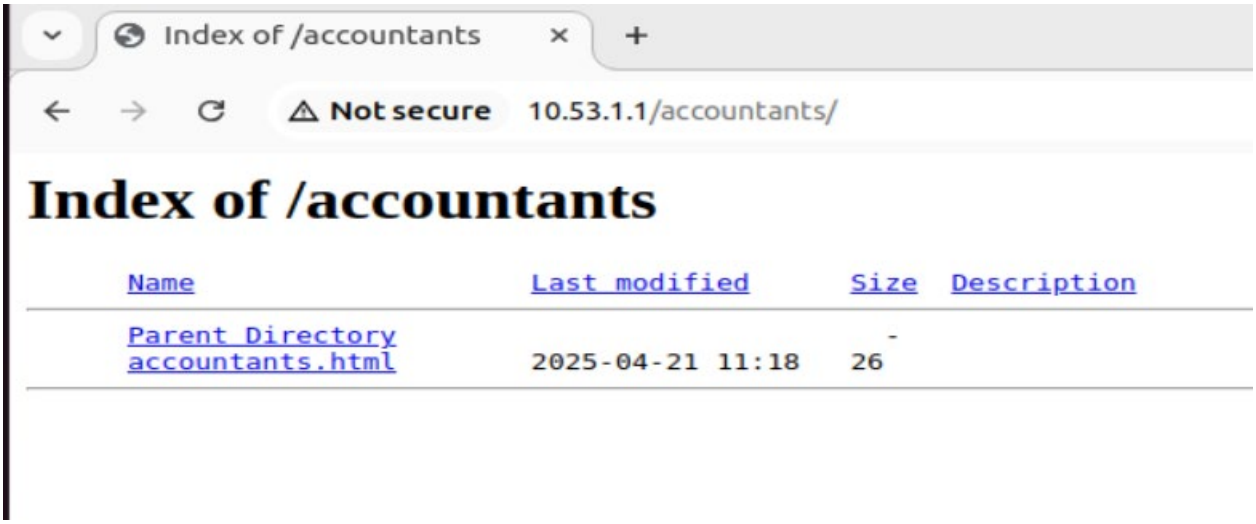


If you select **accountants.html** the following screen is displayed

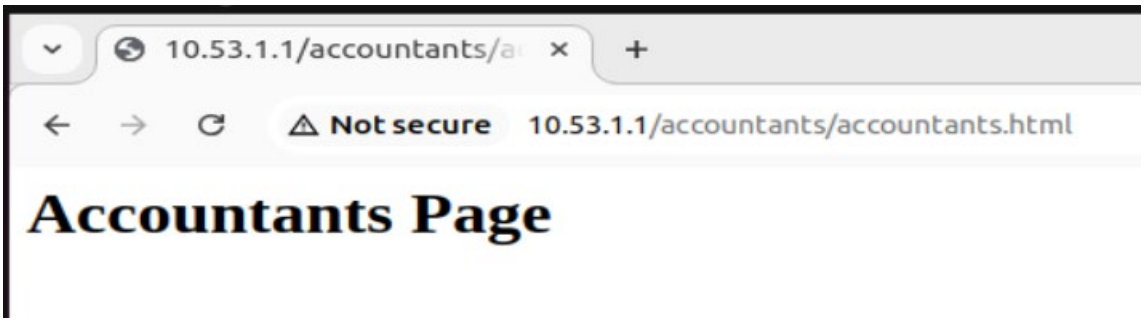


6.4.2.2.4 - Accessible to Programmers (10.53.1.0/24) but not *.gif or *.jpg files

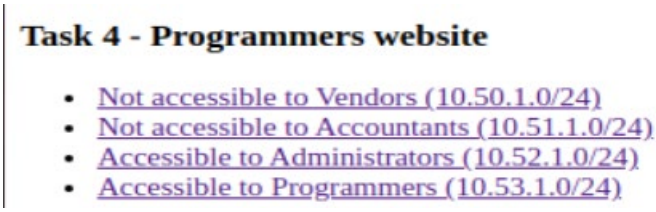
Select fourth choice in the menu and double click to get the next screen.



If you select **accountants.html** the following screen is displayed

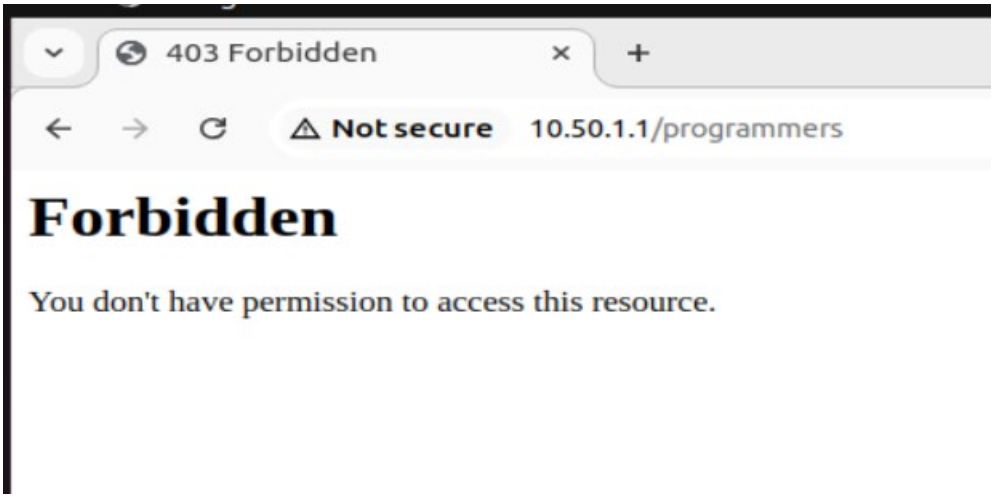


6.4.2.3 Task 4 – Programmers website testing



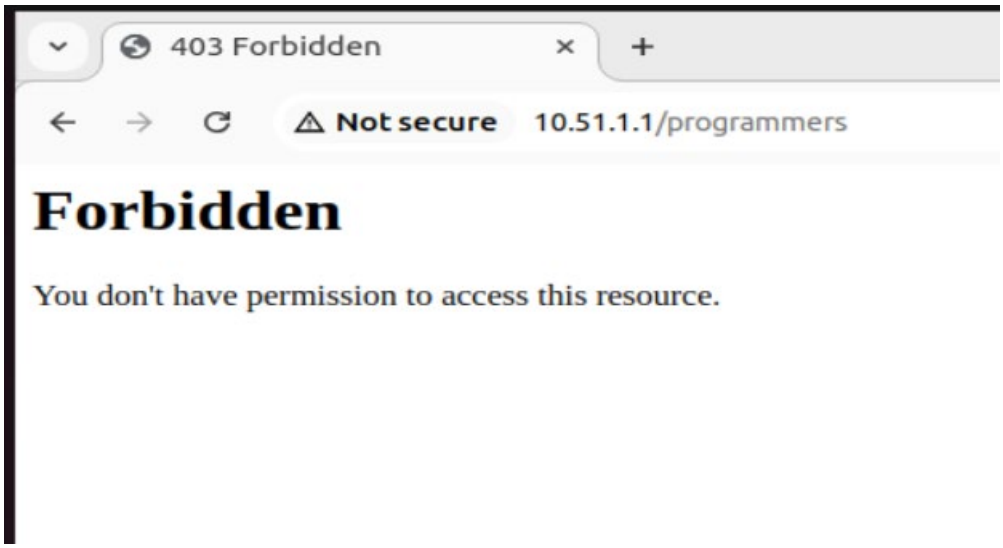
6.4.2.3.1 Not accessible to Vendors (10.50.1.0/24)

Select first choice in the menu and double click to get the next screen.



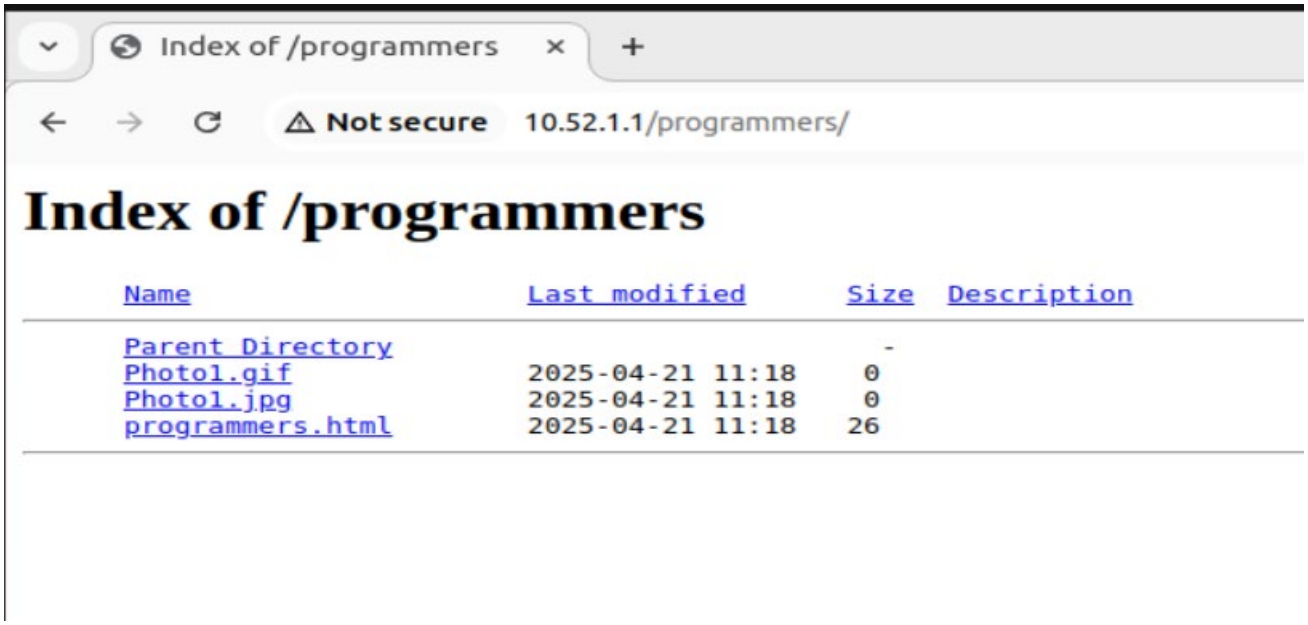
6.4.2.3.2 - Not accessible to Accountants (10.51.1.0/24)

Select second choice in the menu and double click to get the next screen.

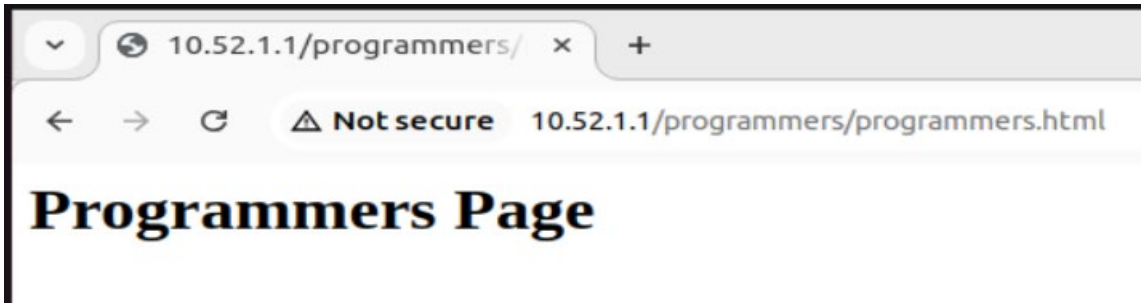


6.4.2.3.3 - Accessible to Administrators (10.52.1.0/24)

Select third choice in the menu and double click to get the next screen.

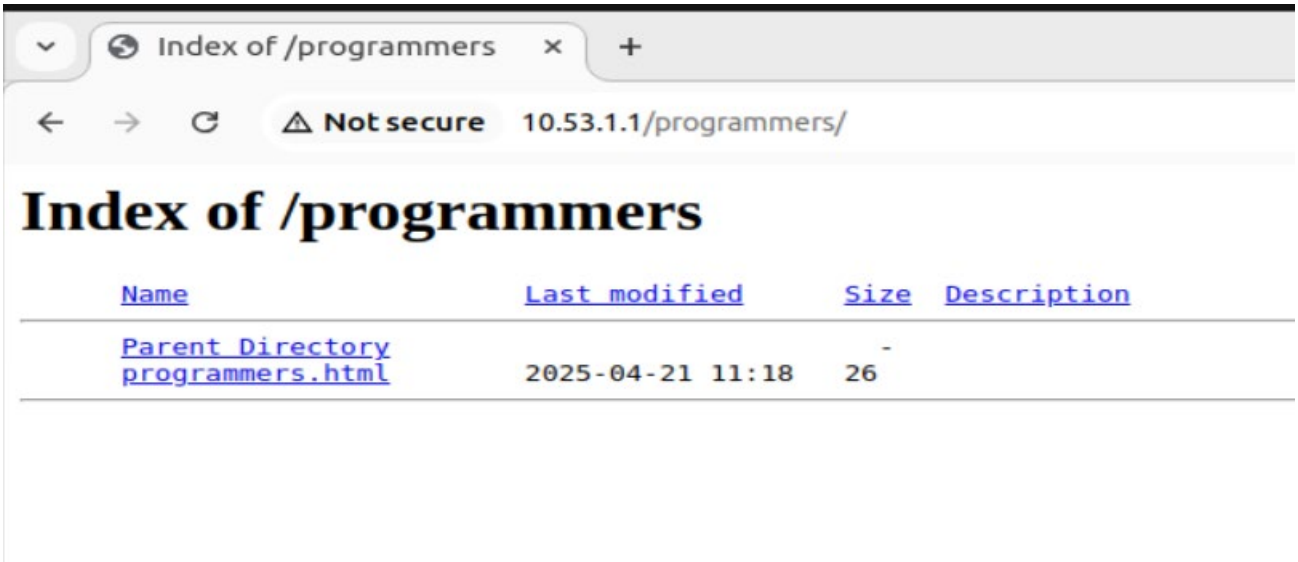


If you select **programmers.html** the following screen is displayed

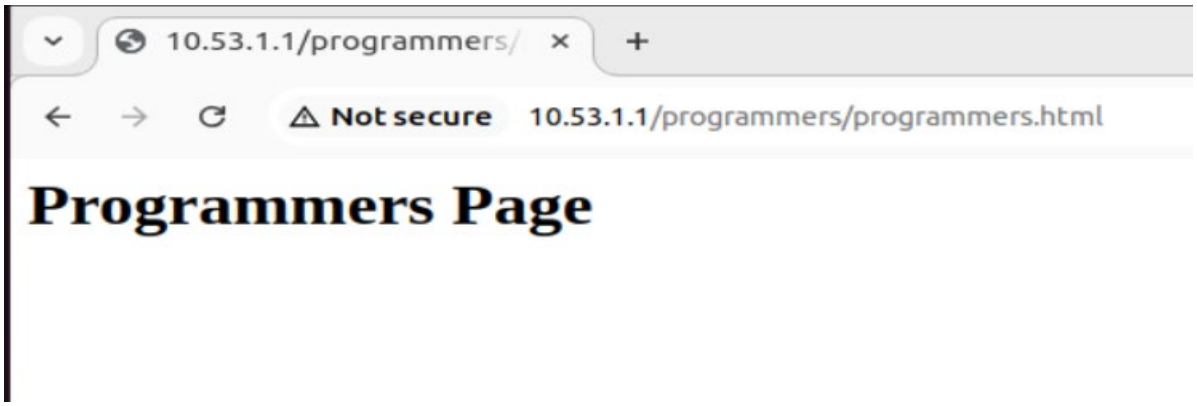


6.4.2.3.4 - Accessible to Programmers (10.53.1.0/24)

Select fourth choice in the menu and double click to get the next screen.



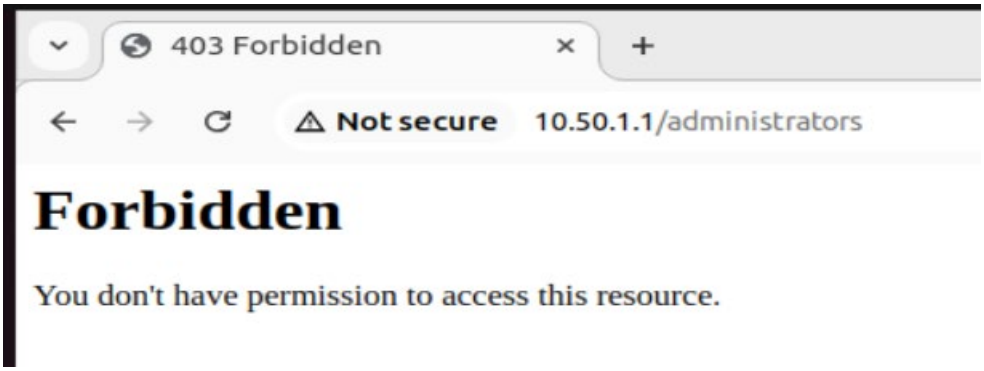
If you select **programmers.html** the following screen is displayed



6.4.2.4 Task 4 – Administrators website testing

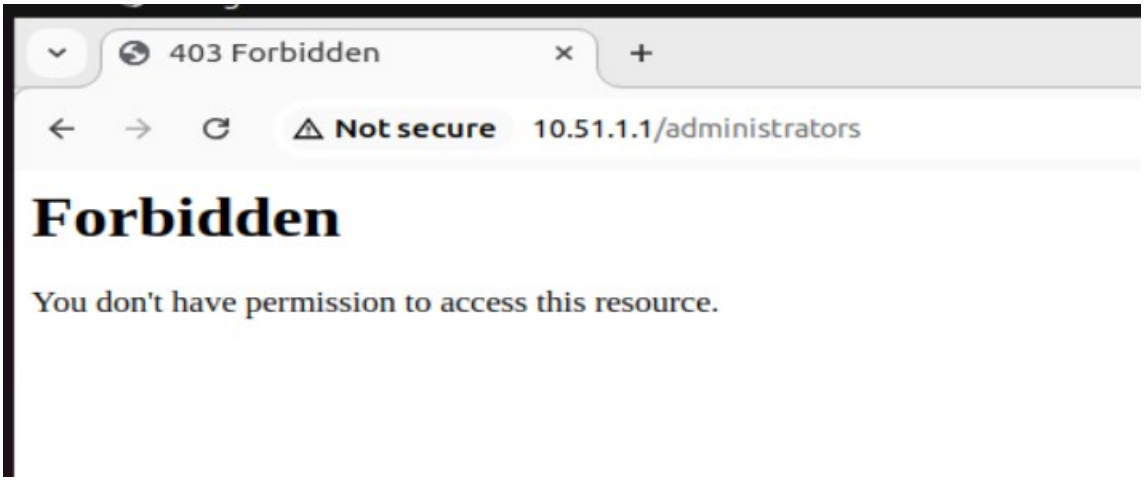
6.4.2.4.1 - Not accessible to Vendors (10.50.1.0/24)

Select first choice in the menu and double click to get the next screen.



6.4.2.4.2 - Not accessible to Accountants (10.51.1.0/24)

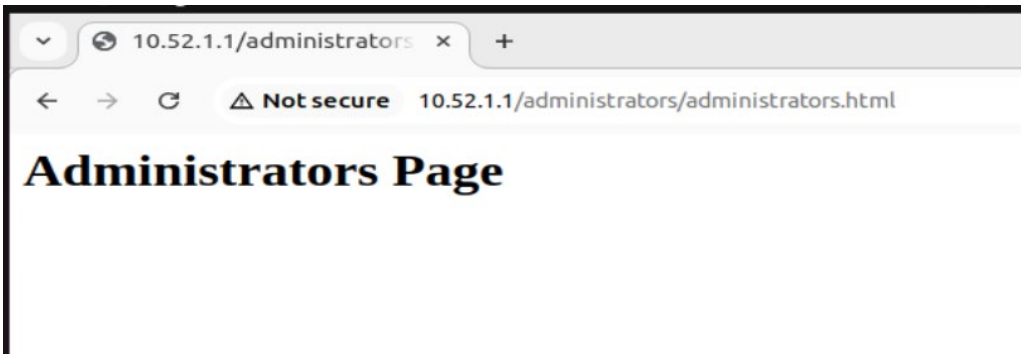
Select second choice in the menu and double click to get the next screen.



6.4.2.4.3 - Accessible to Administrators (10.52.1.0/24)

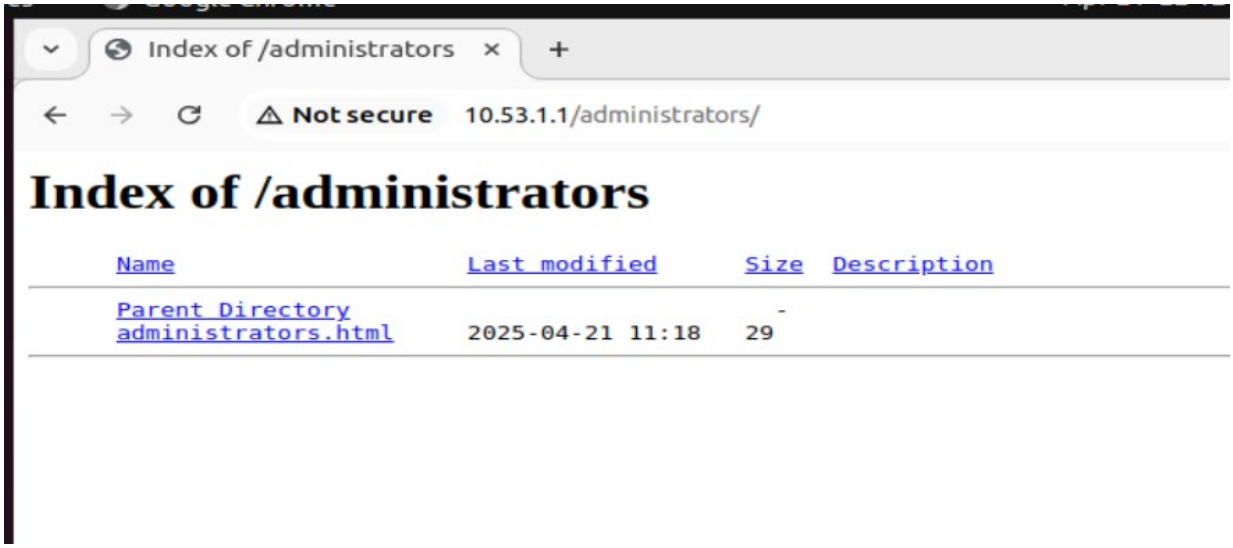
Select third choice in the menu and double click to get the next screen.

If you select **administrators.html** the following screen is displayed

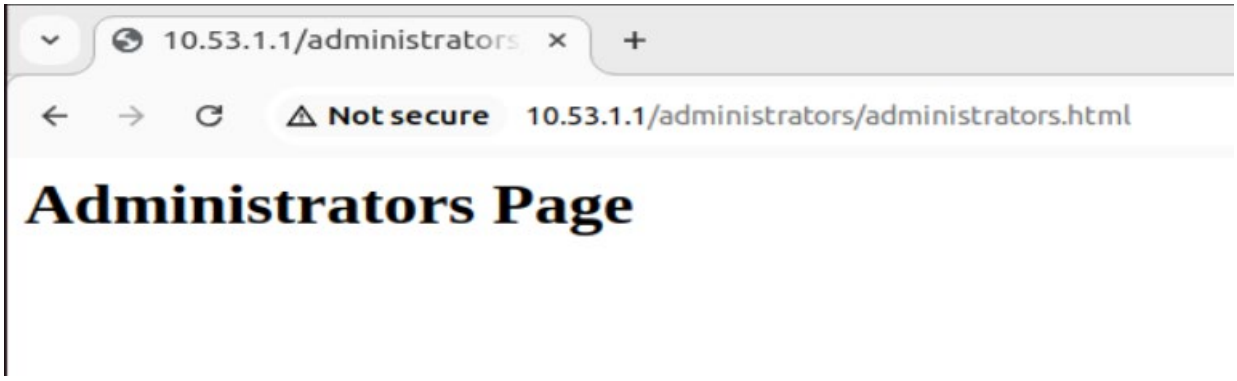


6.4.2.4.4 - Accessible to Programmers (10.53.1.0/24) but not *.gif or *.jpg_files

Select fourth choice in the menu and double click to get the next screen.



If you select **administrators.html** the following screen is displayed



7 Compress Configuration files

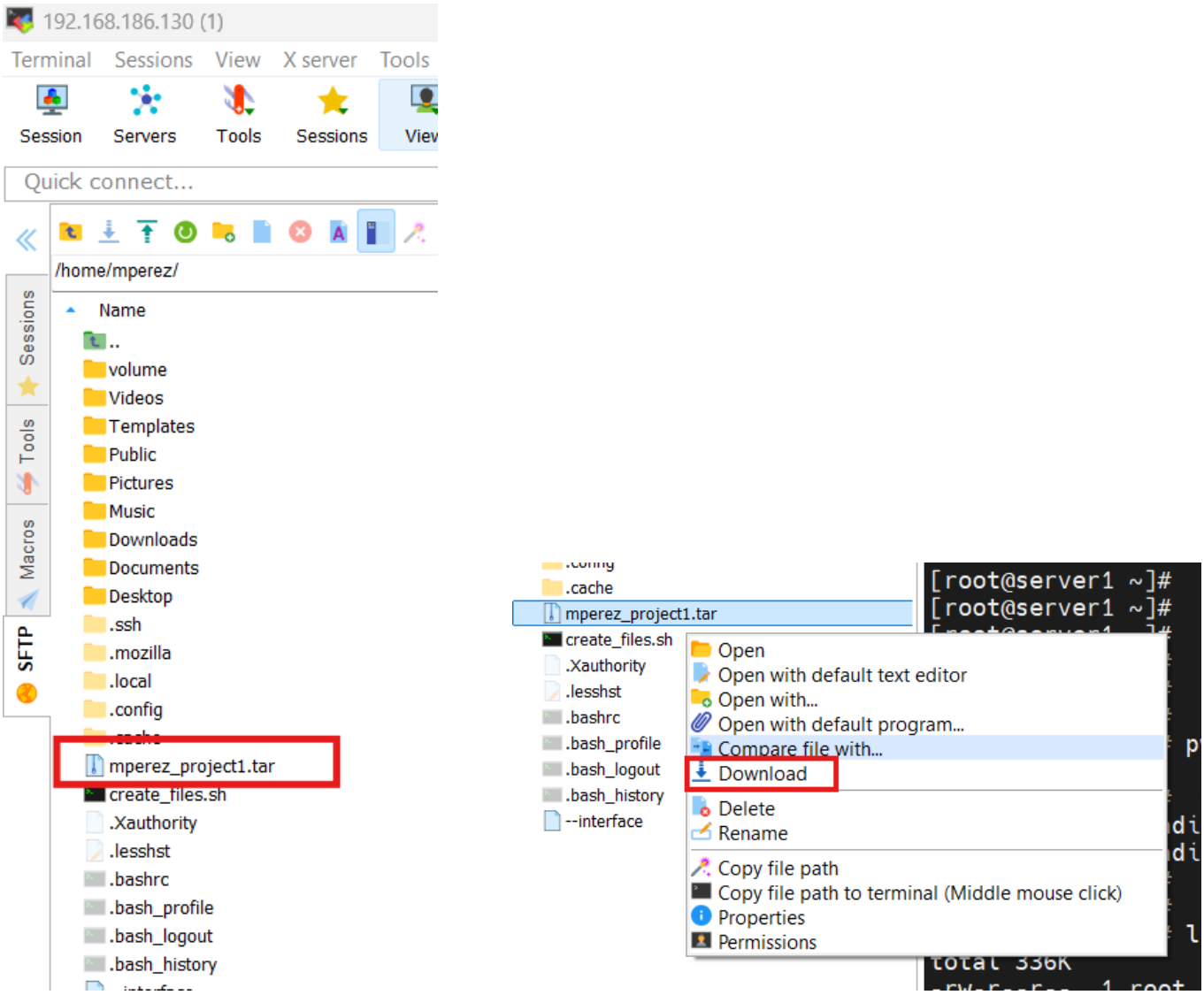
As root, create the compressed file containing of the /etc/httpd/conf/httpd.conf file and containing of /var/www/html_project1 and /var/www/htdocs directories.

tar -czf mperez_project1.tar /etc/httpd/conf/httpd.conf /var/www/html_project1 /var/www/htdocs

Once compressed copy to /home/mperes to be read via sftp

```
/root
[root@server1 ~]#  tar -czf mperez_project1.tar /etc/httpd/conf/httpd.conf /var/www/html_project1  /var/www/htdocs
tar: Removing leading `/' from member names
tar: Removing leading `/' from hard link targets
[root@server1 ~]#
[root@server1 ~]#
[root@server1 ~]# ls -ltrh
total 336K
-rw-r--r--. 1 root root 295K Oct  3  2024 net-tools-2.0-0.64.20160912git.el9.x86_64.rpm
-rw-------. 1 root root    0 Mar 24 14:16 anaconda-ks.cfg
-rw-r--r--. 1 root root  290 Apr 18 16:38 1
-rw-r--r--. 1 root root  34K Apr 21 22:38 mperez_project1.tar
[root@server1 ~]#
[root@server1 ~]#
[root@server1 ~]# cp mperez_project1.tar /home/mperez/
[root@server1 ~]#
```

Download via sftp (Mobaxterm was used)



8 List config files

8.1 List index.html

```
[root@server1 ~]# cat /var/www/html_project1/index.html
<!DOCTYPE html>
<html>
<head>
  <title>Project Part I - Homepage</title>
</head>
<body>
  <p><b><i><u>Welcome to Project Part I</u></i></b></p>
  <hr>
```

```
  <!-- Task 2 - Secure Directories -->
  <h2>Task 2 - Secure Directories</h2>
  <ul>
    <li><a href="http://192.168.50.10/secure1/index.html">Secure1 (user01 - Accessible)</a></li>
    <li><a href="http://192.168.50.10/secure2/index.html">Secure2 (user01 and 192.168.50.0/24 -
Accessible)</a></li>
    <li><a href="http://10.35.16.1/secure2/index.html">Secure2 (user01 and 10.35.16.1/24 - Not
Accessible)</a></li>
    <li><a href="http://192.168.50.10/secure3/index.html">Secure3 (user01 or 192.168.50.0/24 -
Accessible)</a></li>
    <li><a href="http://10.35.16.1/secure3/index.html">Secure3 (user01 or 10.35.16.1/24 - Not
Accessible)</a></li>
    <li><a href="http://192.168.50.10/secure4/index.html">Secure4 (user02 - Accessible)</a></li>
    <li><a href="http://192.168.50.10/secure5/index.html">Secure5 (user01 with .htaccess -
Accessible)</a></li>
    <li><a href="http://192.168.50.10/secure6/index.html">Secure6 (user01 and 192.168.50.0/24 with
.htaccess - Accessible)</a></li>
    <li><a href="http://10.35.16.1/secure6/index.html">Secure6 (user01 and 10.35.16.1/24 with
.htaccess - Not Accessible)</a></li>
    <li><a href="http://192.168.50.10/secure7/index.html">Secure7 (user01 or 192.168.50.0/24 with
.htaccess - Accessible)</a></li>
    <li><a href="http://10.35.16.1/secure7/index.html">Secure7 (user01 or 10.35.16.1/24 with
.htaccess - Accessible)</a></li>
  </ul>

  <!-- Task 3 - Projects -->
  <h2>Task 3 - Projects</h2>

  <!-- Task 3 - Project 1 -->
  <h3>Task 3 - Project 1</h3>
  <ul>
    <li><a href="http://192.168.50.10/Project1/">Project1 (192.168.50.10 - All is accessible)</a></li>
    <li><a href="http://10.35.16.1/Project1/">Project1 (10.35.16.1 accessible except files secret.* and
*.txt)</a></li>
    <li><a href="http://10.35.17.1/Project1/">Project1 (10.35.17.1 accessible except files secret.* and
*.txt)</a></li>
    <li><a href="http://192.168.100.1/Project1/">Project1 (192.168.100.1 Not Accessible)</a></li>
  </ul>

  <!-- Task 3 - Project 2 -->
  <h3>Task 3 - Project 2</h3>
  <ul>
    <li><a href="http://192.168.50.10/Project2/">Project2 (192.168.50.10 - All is accessible)</a></li>
    <li><a href="http://10.35.16.1/Project2/">Project2 (10.35.16.1 Not Accessible)</a></li>
    <li><a href="http://10.35.17.1/Project2/">Project2 (10.35.17.1 accessible except files
*.txt)</a></li>
    <li><a href="http://192.168.100.1/Project2/">Project2 (192.168.100.1 accessible except files
*.txt)</a></li>
```

```
  </ul>

  <!-- Task 3 - Project 3 -->
  <h3>Task 3 - Project 3</h3>
  <ul>
    <li><a href="http://192.168.50.10/Project3/">Project3 (192.168.50.10 - All is accessible)</a></li>
    <li><a href="http://10.35.16.1/Project3/">Project3 (10.35.16.1 Not Accessible)</a></li>
    <li><a href="http://10.35.17.1/Project3/">Project3 (10.35.17.1 Not Accessible)</a></li>
    <li><a href="http://192.168.100.1/Project3/">Project3 (192.168.100.1 accessible except files *.gif
and *.txt)</a></li>
  </ul>

  <!-- Task 3 - Project 4 -->
  <h3>Task 3 - Project 4</h3>
  <ul>
    <li><a href="http://192.168.50.10/Project4/">Project4 (192.168.50.10 - All is accessible)</a></li>
    <li><a href="http://10.35.16.1/Project4/">Project4 (10.35.16.1 accessible except files
test.html)</a></li>
    <li><a href="http://10.35.17.1/Project4/">Project4 (10.35.17.1 Not Accessible)</a></li>
    <li><a href="http://192.168.100.1/Project4/">Project4 (192.168.100.1 accessible except files
test.html)</a></li>
  </ul>


 <!-- Task 4 - -->
  <h2>Task 4</h2>

  <!-- Task 4 - Vendors website -->
  <h3>Task 4 - Vendors website</h3>
  <ul>
    <li><a href="http://10.50.1.1/vendors">Accessible to Vendors (10.50.1.0/24))</a></li>
    <li><a href="http://10.51.1.1/vendors">Not accessible to Accountants (10.51.1.0/24)</a></li>
    <li><a href="http://10.52.1.1/vendors">Accessible to Administrators (10.52.1.0/24)</a></li>
    <li><a href="http://10.53.1.1/vendors">Accessible to Programmers (10.53.1.0/24) but not *.gif or
*.jpg files</a></li>
  </ul>

  <!-- Task 4 - Accountants -->
  <h3>Task 4 - Accountants website</h3>
  <ul>
    <li><a href="http://10.50.1.1/accountants">Not accessible to Vendors (10.50.1.0/24) but not *.html
files</a></li>
    <li><a href="http://10.51.1.1/accountants">Accessible to Accountants (10.51.1.0/24)</a></li>
    <li><a href="http://10.52.1.1/accountants">Accessible to Administrators (10.52.1.0/24)</a></li>
    <li><a href="http://10.53.1.1/accountants">Accessible to Programmers (10.53.1.0/24) but not *.gif
or *.jpg files</a></li>
  </ul>

  <!-- Task 4 - Programmers -->
```

```
  <h3>Task 4 - Programmers website</h3>
  <ul>
    <li><a href="http://10.50.1.1/programmers">Not accessible to Vendors (10.50.1.0/24)</a></li>
    <li><a href="http://10.51.1.1/programmers">Not accessible to Accountants
(10.51.1.0/24)</a></li>
    <li><a href="http://10.52.1.1/programmers">Accessible to Administrators (10.52.1.0/24)</a></li>
    <li><a href="http://10.53.1.1/programmers">Accessible to Programmers (10.53.1.0/24)</a></li>
  </ul>

  <!-- Task 4 - Administrators -->
  <h3>Task 4 - Administrators website</h3>
  <ul>
    <li><a href="http://10.50.1.1/administrators">Not accessible to Vendors (10.50.1.0/24)</a></li>
    <li><a href="http://10.51.1.1/administrators">Not accessible to Accountants
(10.51.1.0/24)</a></li>
    <li><a href="http://10.52.1.1/administrators">Accessible to Administrators (10.52.1.0/24)</a></li>
    <li><a href="http://10.53.1.1/administrators">Accessible to Programmers (10.53.1.0/24) but not
*.gif or *.jpg files</a></li>
  </ul>

  <img src="example.jpg" alt="Example Image"><br>
</body>
</html>
[root@server1 ~]#
[root@server1 ~]#
```

8.2 List httpd.conf

```
[mperez@server1 ~]$ cat /etc/httpd/conf/httpd.conf
ServerRoot "/etc/httpd"
Listen 80
Include conf.modules.d/*.conf

User apache
Group apache

#ServerAdmin root@localhost
ServerName 192.168.50.10

<Directory />
  AllowOverride none
  Require all denied
</Directory>
```

```
DocumentRoot "/var/www/html_project1"

<Directory "/var/www">
  AllowOverride None
  Require all granted
</Directory>

<Directory "/var/www/html_project1">
  Options Indexes FollowSymLinks
  AllowOverride All
  Require all granted
</Directory>

<Directory "/var/www/html_project1/secure1">
  AuthType Basic
  AuthName "Restricted Access - secure1"
  AuthUserFile /etc/httpd/.htpasswd
  Require user user01
</Directory>

<Directory "/var/www/html_project1/secure2">
  AuthType Basic
  AuthName "Restricted Access - secure2"
  AuthUserFile /etc/httpd/.htpasswd
  <RequireAll>
    Require user user01
    Require ip 192.168.50.0/24
  </RequireAll>
</Directory>

<Directory "/var/www/html_project1/secure3">
  AuthType Basic
  AuthName "Restricted Access - secure3"
  AuthUserFile /etc/httpd/.htpasswd
  <RequireAny>
    Require user user01
    Require ip 192.168.50.0/24
  </RequireAny>
</Directory>

<Directory "/var/www/html_project1/secure4">
  AuthType Basic
  AuthName "Restricted Access - secure4"
  AuthUserFile /etc/httpd/.htpasswd
  Require user user02
</Directory>

<Directory "/var/www/html_project1/secure5">
  AllowOverride All
```

```
</Directory>

<Directory "/var/www/html_project1/secure6">
  AllowOverride All
</Directory>

<Directory "/var/www/html_project1/secure7">
  AllowOverride All
</Directory>

# Project directories configuration

# Project1 Configuration
<Directory "/var/www/html_project1/Project1">
  Options +Indexes
  IndexOptions FancyIndexing
  AllowOverride None
  <RequireAny>
    Require ip 10.35.16.0/24
    Require ip 10.35.17.0/24
    Require ip 192.168.50.0/24
  </RequireAny>
  <Files "secret.*">
    Require all denied
  </Files>
</Directory>

# Project2 Configuration
<Directory "/var/www/html_project1/Project2">
  Options +Indexes
  IndexOptions FancyIndexing
  AllowOverride None
  <RequireAll>
    Require all granted
    Require not ip 10.35.16.0/24
  </RequireAll>
</Directory>

# Project3 Configuration
<Directory "/var/www/html_project1/Project3">
  Options +Indexes
  IndexOptions FancyIndexing
  AllowOverride None
  <RequireAny>
    Require ip 192.168.100.0/24
    Require ip 192.168.50.0/24
  </RequireAny>
  <Files "*.gif">
    Require all denied
```

```
  </Files>
</Directory>

# Project4 Configuration
<Directory "/var/www/html_project1/Project4">
  Options +Indexes
  IndexOptions FancyIndexing
  AllowOverride None
  <RequireAny>
    Require ip 10.35.16.0/24
    Require ip 192.168.100.0/24
    Require ip 192.168.50.0/24
  </RequireAny>
  <Files "test.html">
    Require all denied
  </Files>
</Directory>

# Global restriction for all .txt files across all directories
<Files "*.txt">
  Require all denied
</Files>

# Task 4 - Authorization Configuration

# Ensure parent directory access for aliases
<Directory "/var/www">
  Options None
  AllowOverride None
  Require all granted
</Directory>

# Allow access to Task 4 directories by default
<Directory "/var/www/htdocs">
  Options Indexes
  AllowOverride None
</Directory>

# Define aliases for Task 4 directories
Alias /vendors "/var/www/htdocs/vendors"
Alias /accountants "/var/www/htdocs/accountants"
Alias /administrators "/var/www/htdocs/administrators"
Alias /programmers "/var/www/htdocs/programmers"

# Vendors Directory
<Directory "/var/www/htdocs/vendors">
  Options +Indexes
  IndexOptions FancyIndexing
  AllowOverride None
```

```
    <RequireAny>
      Require ip 10.50.1.0/24
      Require ip 10.52.1.0/24
      Require ip 10.53.1.0/24
    </RequireAny>
    # Deny *.gif and *.jpg for Programmers in this directory
    <FilesMatch "\.(gif|jpg)$">
      <RequireAll>
        Require all granted
        Require not ip 10.53.1.0/24
      </RequireAll>
    </FilesMatch>
</Directory>

<Directory "/var/www/htdocs/accountants">
    Options +Indexes
    IndexOptions FancyIndexing
    AllowOverride None

    # Allow only Accountants, Administrators, and Programmers
    <RequireAny>
      Require ip 10.51.1.0/24
      Require ip 10.52.1.0/24
      Require ip 10.53.1.0/24
    </RequireAny>

    # Block Accountants from HTML files
    <FilesMatch "\.html$">
      <RequireAll>
        Require all granted
        Require not ip 10.51.1.0/24
      </RequireAll>
    </FilesMatch>

    # Block Programmers from image files
    <FilesMatch "\.(gif|jpg)$">
      <RequireAll>
        Require all granted
        Require not ip 10.53.1.0/24
      </RequireAll>
    </FilesMatch>

    DirectoryIndex disabled
</Directory>

# Administrators Directory
<Directory "/var/www/htdocs/administrators">
    Options +Indexes
    IndexOptions FancyIndexing
```

```
  AllowOverride None
  <RequireAny>
    Require ip 10.52.1.0/24
    Require ip 10.53.1.0/24
  </RequireAny>
  # Deny *.gif and *.jpg for Programmers in this directory
  <FilesMatch "\.(gif|jpg)$">
    <RequireAll>
      Require all granted
      Require not ip 10.53.1.0/24
    </RequireAll>
  </FilesMatch>
</Directory>

# Programmers Directory
<Directory "/var/www/htdocs/programmers">
  Options +Indexes
  IndexOptions FancyIndexing
  AllowOverride None
  <RequireAny>
    Require ip 10.52.1.0/24
    Require ip 10.53.1.0/24
  </RequireAny>
  # Deny *.gif and *.jpg for Programmers in this directory
  <FilesMatch "\.(gif|jpg)$">
    <RequireAll>
      Require all granted
      Require not ip 10.53.1.0/24
    </RequireAll>
  </FilesMatch>
</Directory>

<IfModule dir_module>
  DirectoryIndex index.html
</IfModule>

#
# The following lines prevent .htaccess and .htpasswd files from being
# viewed by Web clients.
#
<Files ".ht*">
  Require all denied
</Files>

#
# ErrorLog: The location of the error log file.
# If you do not specify an ErrorLog directive within a <VirtualHost>
# container, error messages relating to that virtual host will be
# logged here.  If you *do* define an error logfile for a <VirtualHost>
```

```
# container, that host's errors will be logged there and not here.
#
ErrorLog "logs/error_log"

#
# LogLevel: Control the number of messages logged to the error_log.
# Possible values include: debug, info, notice, warn, error, crit,
# alert, emerg.
#
LogLevel warn

<IfModule log_config_module>
    #
    # The following directives define some format nicknames for use with
    # a CustomLog directive (see below).
    #
    LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined
    LogFormat "%h %l %u %t \"%r\" %>s %b" common

    <IfModule logio_module>
      # You need to enable mod_logio.c to use %I and %O
      LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\" %I %O" combinedio
    </IfModule>

    #
    # The location and format of the access logfile (Common Logfile Format).
    # If you do not define any access logfiles within a <VirtualHost>
    # container, they will be logged here.  Contrariwise, if you *do*
    # define per-<VirtualHost> access logfiles, transactions will be
    # logged therein and *not* in this file.
    #
    #CustomLog "logs/access_log" common

    #
    # If you prefer a logfile with access, agent, and referer information
    # (Combined Logfile Format) you can use the following directive.
    #
    CustomLog "logs/access_log" combined
</IfModule>

<IfModule alias_module>
    #
    # Redirect: Allows you to tell clients about documents that used to
    # exist in your server's namespace, but do not anymore. The client
    # will make a new request for the document at its new location.
    # Example:
    # Redirect permanent /foo http://www.example.com/bar

    #
```

```
    # Alias: Maps web paths into filesystem paths and is used to
    # access content that does not live under the DocumentRoot.
    # Example:
    # Alias /webpath /full/filesystem/path
    #
    # If you include a trailing / on /webpath then the server will
    # require it to be present in the URL.  You will also likely
    # need to provide a <Directory> section to allow access to
    # the filesystem path.

    #
    # ScriptAlias: This controls which directories contain server scripts.
    # ScriptAliases are essentially the same as Aliases, except that
    # documents in the target directory are treated as applications and
    # run by the server when requested rather than as documents sent to the
    # client.  The same rules about trailing "/" apply to ScriptAlias
    # directives as to Alias.
    #
    ScriptAlias /cgi-bin/ "/var/www/cgi-bin/"

</IfModule>

#
# "/var/www/cgi-bin" should be changed to whatever your ScriptAliased
# CGI directory exists, if you have that configured.
#
<Directory "/var/www/cgi-bin">
    AllowOverride None
    Options None
    Require all granted
</Directory>

<IfModule mime_module>
    #
    # TypesConfig points to the file containing the list of mappings from
    # filename extension to MIME-type.
    #
    TypesConfig /etc/mime.types

    #
    # AddType allows you to add to or override the MIME configuration
    # file specified in TypesConfig for specific file types.
    #
    #AddType application/x-gzip .tgz
    #
    # AddEncoding allows you to have certain browsers uncompress
    # information on the fly. Note: Not all browsers support this.
    #
    #AddEncoding x-compress .Z
```

```
  #AddEncoding x-gzip .gz .tgz
  #
  # If the AddEncoding directives above are commented-out, then you
  # probably should define those extensions to indicate media types:
  #
  AddType application/x-compress .Z
  AddType application/x-gzip .gz .tgz

  #
  # AddHandler allows you to map certain file extensions to "handlers":
  # actions unrelated to filetype. These can be either built into the server
  # or added with the Action directive (see below)
  #
  # To use CGI scripts outside of ScriptAliased directories:
  # (You will also need to add "ExecCGI" to the "Options" directive.)
  #
  #AddHandler cgi-script .cgi

  # For type maps (negotiated resources):
  #AddHandler type-map var

  #
  # Filters allow you to process content before it is sent to the client.
  #
  # To parse .shtml files for server-side includes (SSI):
  # (You will also need to add "Includes" to the "Options" directive.)
  #
  AddType text/html .shtml
  AddOutputFilter INCLUDES .shtml
</IfModule>

#
# Specify a default charset for all content served; this enables
# interpretation of all content as UTF-8 by default.  To use the
# default browser choice (ISO-8859-1), or to allow the META tags
# in HTML content to override this choice, comment out this
# directive:
#
AddDefaultCharset UTF-8

<IfModule mime_magic_module>
  #
  # The mod_mime_magic module allows the server to use various hints from the
  # contents of the file itself to determine its type.  The MIMEMagicFile
  # directive tells the module where the hint definitions are located.
  #
  MIMEMagicFile conf/magic
</IfModule>
```

```
#
# Customizable error responses come in three flavors:
# 1) plain text 2) local redirects 3) external redirects
#
# Some examples:
#ErrorDocument 500 "The server made a boo boo."
#ErrorDocument 404 /missing.html
#ErrorDocument 404 "/cgi-bin/missing_handler.pl"
#ErrorDocument 402 http://www.example.com/subscription_info.html
ServerRoot "/etc/httpd"
[mperez@server1 ~]$
```