Web Server on KdG LAN Documentation
Sprint 1

Server IP address: 10.134.178.2
Team: Team 02

Name	Task No.
Vorozhtsova Anna	2,3,4,6
Al Marchouhe Bilal	2
Buinceanu Vlad	1,2,5,6,7
Ramoska Emilis	8

1. After setting up the GlobalProtect vpn the server could be accessed via telnet using the following commands

```
vladbuinceanu@Vlads-MacBook-Air ~ % telnet 10.134.178.2
Trying 10.134.178.2...
Connected to 10.134.178.2.
Escape character is '^]'.
Password:
Login incorrect
02int1 login:
Password:
Last login: Wed Nov 23 22:12:08 from 10.52.194.73
[team@team02int1 ~]$
```

2. Enable SSH service
   Enabled the SSH service using the following command

```
[team@team02int1 ~]$ sudo systemctl start sshd
```

   Made sure the SSH service start at system boot using the following command

```
[team@team02int1 ~]$ sudo systemctl enable sshd
```

3. Created accounts for members and coaches:
   Created a username and a default password for every member and coach

Full Name	**Username**	**Password**
Toni Mini	toni	initial01
Piet Boedt	piet	initial01
Al Marchouhe Bilal	bilal	initial01
Oussama Ez-Zayyany	osu	initial01
Vorozhtsova Anna	anna	initial01
Buinceanu Vlad	vlad	initial01
Ramoska Emilis	emilis	Initial01

Using the following commands:

```
[team@team02int1 ~]$ sudo useradd anna
[team@team02int1 ~]$ sudo useradd vlad
[team@team02int1 ~]$ sudo useradd emilis
[…]
```

```
[team@team02int1 ~]$ sudo passwd vlad
[sudo] password for vlad:
Changing password for user vlad.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[team@team02int1 ~]$ sudo passwd bilal
[…]
```

a. Give every user administrator rights by adding them to the wheel group

```
[team@team02int1 ~]$ sudo usermod -aG wheel bilal
[team@team02int1 ~]$ sudo usermod -aG wheel osu
[…]
```

Check if the users have administrator rights

```
[vlad@team02int1 ~]$ sudo -l
Matching Defaults entries for vlad on team02int1:
      !visiblepw, always_set_home, match_group_by_gid,
always_query_group_plugin,
      env_reset, env_keep="COLORS DISPLAY HOSTNAME HISTSIZE KDEDIR
LS_COLORS",
      env_keep+="MAIL PS1 PS2 QTDIR USERNAME LANG LC_ADDRESS
LC_CTYPE",
      env_keep+="LC_COLLATE LC_IDENTIFICATION LC_MEASUREMENT
LC_MESSAGES",
      env_keep+="LC_MONETARY LC_NAME LC_NUMERIC LC_PAPER
LC_TELEPHONE", env_keep+="LC_TIME
      LC_ALL LANGUAGE LINGUAS _XKB_CHARSET XAUTHORITY",
      secure_path=/sbin\:/bin\:/usr/sbin\:/usr/bin

User vlad may run the following commands on team02int1:
      (ALL) ALL
```

b. Every new user is required to change his or her password on first login

```
[team@team02int1 ~]$ sudo chage -d 0 osu
[sudo] password for team:
[team@team02int1~]$ sudo chage -d 0 vlad
[sudo] password for team:
```

4. Created a new group and added all the users to it

```
[anna@team02int1 ~]$ sudo groupadd students
[sudo] password for team:
[anna@team02int1 ~]$ usermod -aG students osu
[…]
```

Check if the user has been added to the needed group

```
[anna@team02int1 ~]$ id osu
uid=2003(osu)gid=2003(osu)groups=2003(osu),10(wheel),991(students)
```

5. Lock the default initial server account

```
[vlad@team02int1 ~]$ sudo usermod -L team
```

6. Configure remote access to the server via SSH using key pairs

   Created a key pair:

```
vladbuinceanu@Vlads-MacBook-Air ~ % ssh-keygen -f
~/.ssh/key-with-pass

Generating public/private rsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:

Your identification has been saved in
/Users/vladbuinceanu/.ssh/key-with-pass

Your public key has been saved in
/Users/vladbuinceanu/.ssh/key-with-pass.pub
```

   Copied the public key of the SSH key pair to the server:

```
vladbuinceanu@Vlads-MacBook-Air ~ % ssh-copy-id vlad@10.134.178.2
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed:
"/Users/vladbuinceanu/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new
key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if
you are prompted now it is to install the new keys
vlad@10.134.178.2's password:

Number of key(s) added:     1

Now try logging into the machine, with:  "ssh 'vlad@10.134.178.2'"
and check to make sure that only the key(s) you wanted were added.
```

Prevented root from login by setting the PermitRootLogin to “no”

```
[vlad@team02int1 ~]$ sudo vim /etc/ssh/sshd_config
[sudo] password for vlad:
```

```
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
PermitRootLogin no
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10
```

7. Disabled telnet service
   Checked if the ssh is running

```
[vlad@team02int1 ~]$ systemctl status sshd
● sshd.service - OpenSSH server daemon
     Loaded: loaded (/usr/lib/systemd/system/sshd.service;
enabled; vendor preset: enable>
     Active: active (running) since Fri 2022-11-25 22:36:17 CET; 2
days ago
       Docs: man:sshd(8)
             man:sshd_config(5)
    Process: 13386 ExecReload=/bin/kill -HUP $MAINPID
(code=exited, status=0/SUCCESS)
   Main PID: 13369 (sshd)
      Tasks: 1 (limit: 23572)
     Memory: 11.1M
        CPU: 3.703s
     CGroup: /system.slice/sshd.service
             └─13369 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100
startups"
```

Disable telnet

```
[vlad@team02int1 ~]$ sudo systemctl disable telnet.socket
[sudo] password for vlad:
Removed /etc/systemd/system/sockets.target.wants/telnet.socket.
```

8. Created a shared folder structure for the scripts in the /opt folder

```
[emilis@team02int1 opt]$ sudo mkdir Scripts
[emilis@team02int1 opt]$ sudo chgrp students Scripts
[emilis@team02int1 opt]$ ls -l
total 0
drwxr-xr-x. 2 root students 6 Nov 24 11:06 Scripts
```

Granted the right permissions

```
[emilis@team02int1 opt]$ sudo chmod 2770 Scripts/
[sudo] password for vlad:
[emlis@team02int1 opt]$ ls -l
total 0
drwxrws---. 2 root students 6 Nov 24 11:06 Scripts
[emilis@team02int1 opt]$ touch Scripts/test
[emilis@team02int1 opt]$ ls -l Scripts/
total 0
-rw-r--r--. 1 vlad students 0 Nov 24 11:18 test
```

Things to do after the feedback:

- Make every folder and username lowercase

Sprint 2

Server IP address: 10.134.178.2
Team: Team 02

Name	Task	Reviewed by
Vorozhtsova Anna	1	Buinceanu Vlad
Al Marchouhe Bilal	2	Buinceanu Vlad
Buinceanu Vlad	3,4,5,6	Vorozhtsova Anna
Ramoska Emilis		
Ez-Zayyany Oussama	2	Buinceanu Vlad

1. Implement the feedback from Sprint 1
   Made all the users lowercase

```
[anna@team02int1 ~]$ tail /etc/passwd
anna:x:2001:2001::/home/anna:/bin/bash
bilal:x:2002:2002::/home/bilal:/bin/bash
osu:x:2003:2003::/home/osu:/bin/bash
vlad:x:2005:2005::/home/vlad:/bin/bash
piet:x:2006:2006::/home/piet:/bin/bash
toni:x:2007:2007::/home/toni:/bin/bash
```

   Made all the folders lowercase

```
[anna@team02int1 opt]$ mv Scripts/ scripts
```

2. Install apache web server

```
[bilal@team02int1 ~]$ sudo dnf install httpd
Last metadata expiration check: 0:10:07 ago on Sat 10 Dec 2022
04:55:10 PM CET.
Dependencies resolved.
==================================================================
 Package                  Arch       Version
Repository      Size
==================================================================
Installing:
 httpd       x86_64   2.4.53-7.el9         appstream       48 k

[...]

Installed:
  almalinux-logos-httpd-90.5.1-1.1.el9.noarch
  httpd-2.4.53-7.el9.x86_64
  httpd-core-2.4.53-7.el9.x86_64
  httpd-filesystem-2.4.53-7.el9.noarch
  mailcap-2.1.49-5.el9.noarch
  mod_http2-1.15.19-2.el9.x86_64
  mod_lua-2.4.53-7.el9.x86_64

Complete!
```

a. Make sure the web server starts starts automatically on boot

```
[bilal@team02int1 ~]$ sudo systemctl enable httpd
Created symlink
/etc/systemd/system/multi-user.target.wants/httpd.service →
/usr/lib/systemd/system/httpd.service.
```

b. Start the webserver

```
[bilal@team02int1 ~]$ sudo systemctl start httpd
```

c. Check if the web server is running

```
[bilal@team02int1 ~]$ sudo systemctl status httpd
● httpd.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/httpd.service;
enabled; vendor pre>
     Active: active (running) since Sat 2022-12-10 17:05:36 CET;
21s ago
       Docs: man:httpd.service(8)
   Main PID: 68035 (httpd)
```

```
[bilal@team02int1 ~]$ systemctl is-enabled httpd
enabled
```

3. Copy the team website to the webserver

a. Install rsync

```
[vlad@team02int1 ~]$ sudo dnf install rsync
[sudo] password for vlad:
Last metadata expiration check: 1:58:50 ago on Mon 12 Dec 2022
09:52:59 AM CET.
Dependencies resolved.
================================================================
 Package         Architecture     Version     Repository     Size
================================================================
Installing:
 rsync            x86_64      3.2.3-18.el9     baseos        393 k

[...]


Transaction test succeeded.
Running transaction
  Preparing        :                                       1/1
  Installing       : rsync-3.2.3-18.el9.x86_64             1/1
  Verifying        : rsync-3.2.3-18.el9.x86_64             1/1

Installed:
  rsync-3.2.3-18.el9.x86_64
```

b. Copy the web site files from the local machine to the webserver

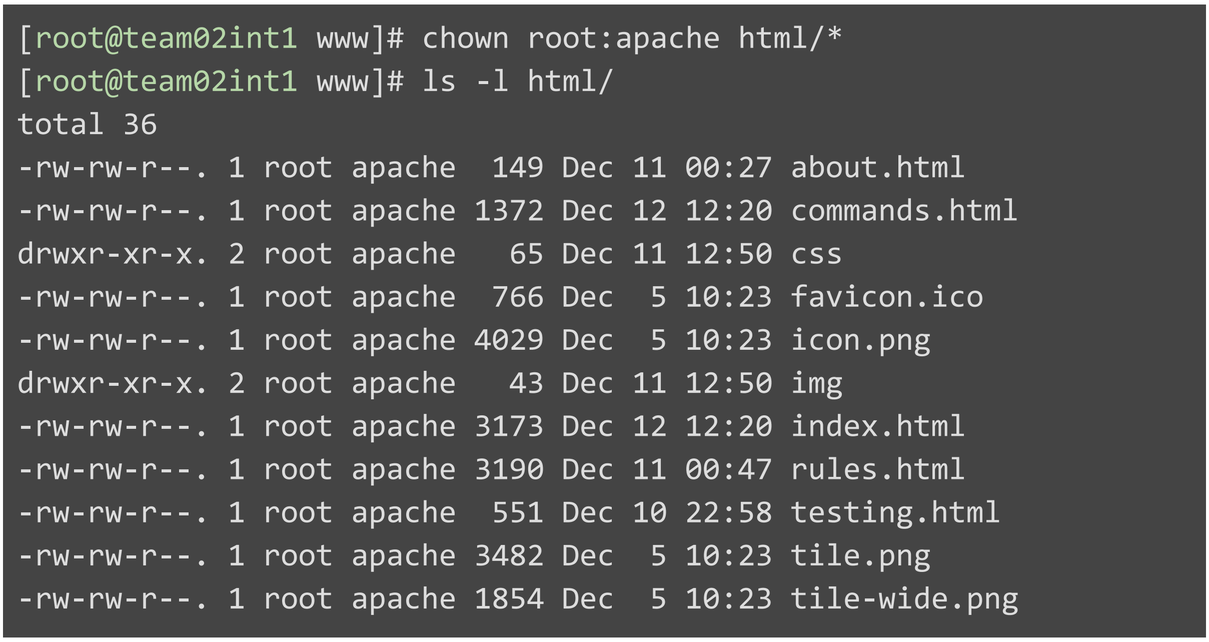
```
➜  ~ rsync -av [...]/KdG/Website vlad@10.134.178.2:/home/vlad
building file list ... done
Website/
Website/.htaccess
Website/404.html
Website/LICENSE.txt
Website/browserconfig.xml

[...]

sent 484326 bytes  received 1042 bytes  323578.67 bytes/sec
total size is 481020  speedup is 0.99
```

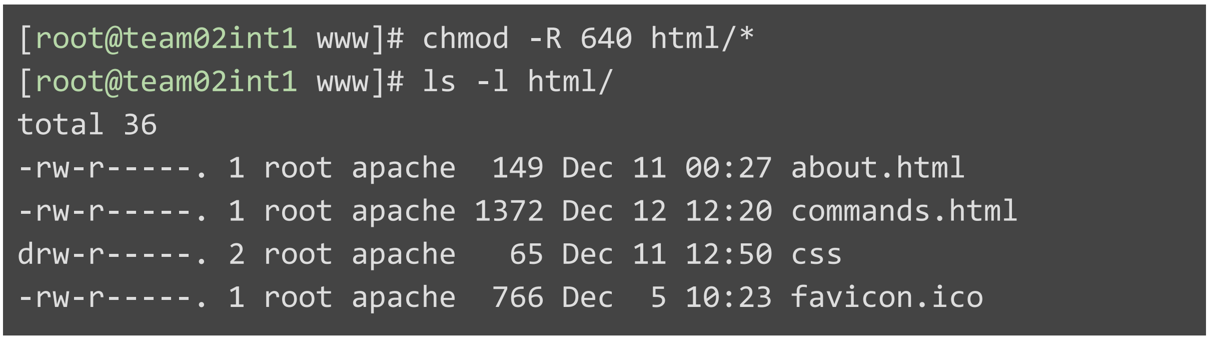
4. Secure the web pages

a. All web server pages should be owned by root and the group ownership should be granted to the group of the user which runs the httpd process



```
[root@team02int1 www]# chown root:apache html/*
[root@team02int1 www]# ls -l html/
total 36
-rw-rw-r--. 1 root apache  149 Dec 11 00:27 about.html
-rw-rw-r--. 1 root apache 1372 Dec 12 12:20 commands.html
drwxr-xr-x. 2 root apache   65 Dec 11 12:50 css
-rw-rw-r--. 1 root apache  766 Dec  5 10:23 favicon.ico
-rw-rw-r--. 1 root apache 4029 Dec  5 10:23 icon.png
drwxr-xr-x. 2 root apache   43 Dec 11 12:50 img
-rw-rw-r--. 1 root apache 3173 Dec 12 12:20 index.html
-rw-rw-r--. 1 root apache 3190 Dec 11 00:47 rules.html
-rw-rw-r--. 1 root apache  551 Dec 10 22:58 testing.html
-rw-rw-r--. 1 root apache 3482 Dec  5 10:23 tile.png
-rw-rw-r--. 1 root apache 1854 Dec  5 10:23 tile-wide.png
```

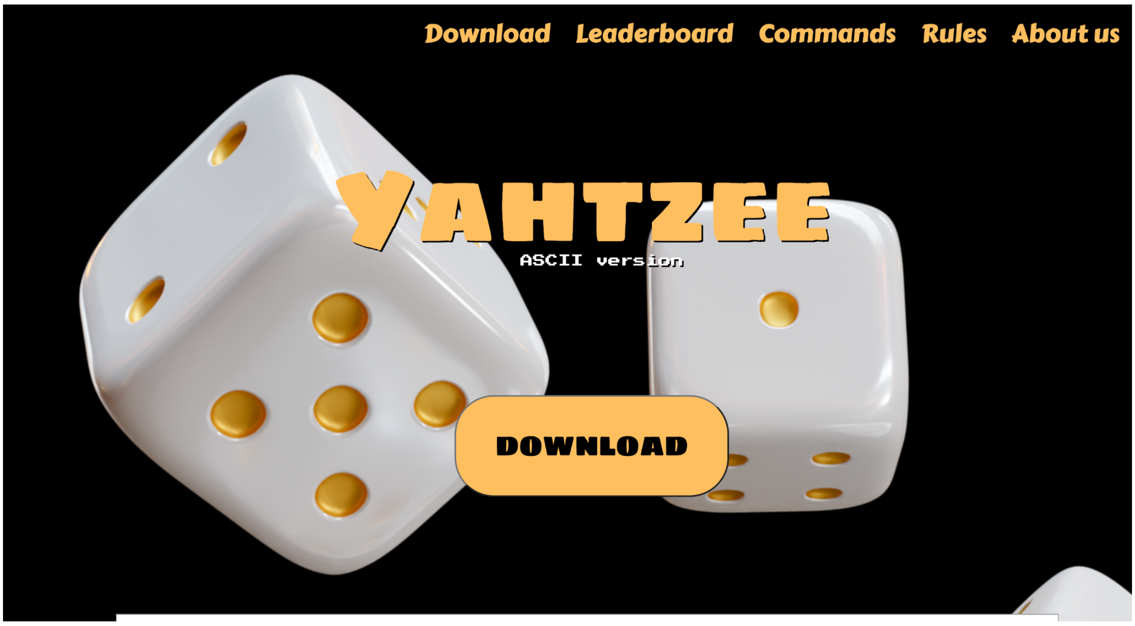
b. Owner can read/write the files, Group can read the files, Others should have no access.

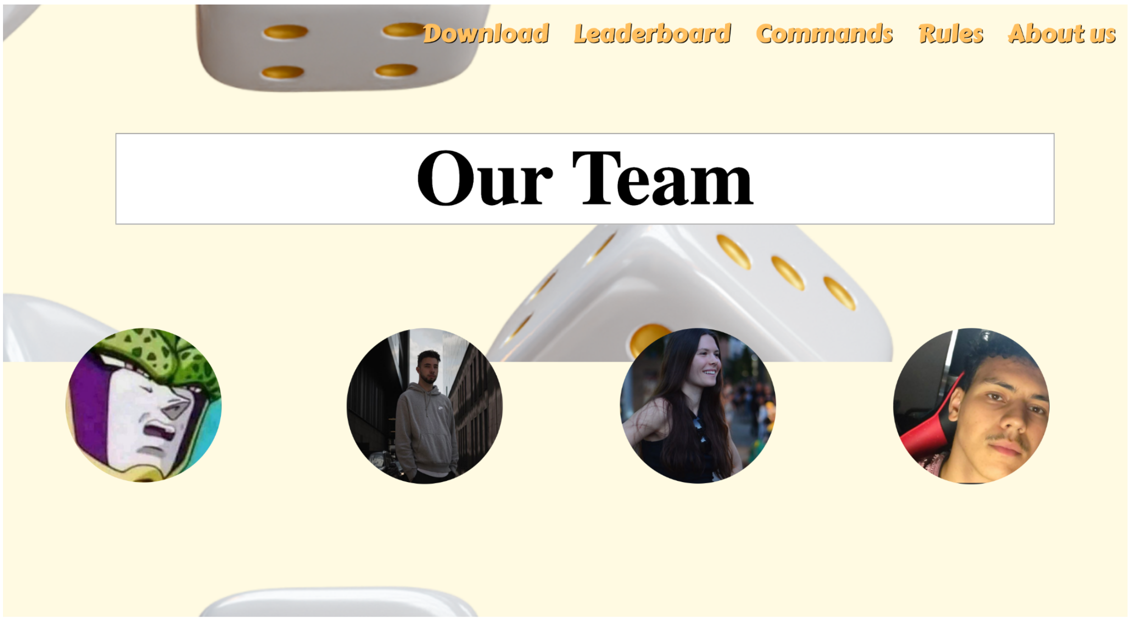


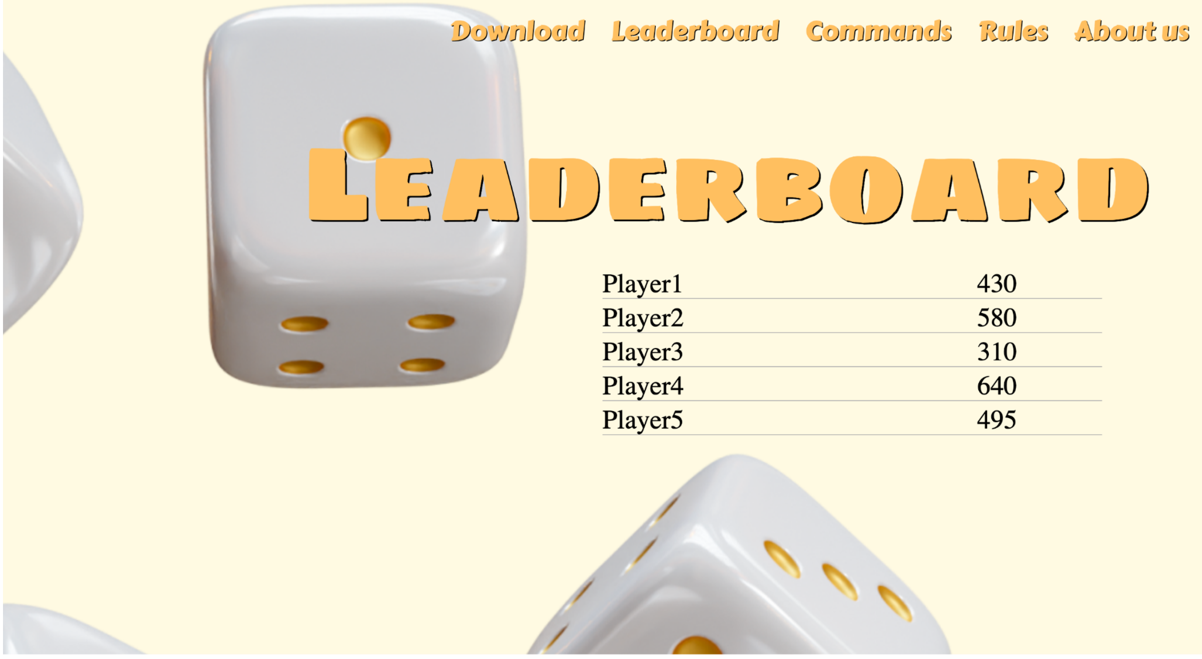
```
[root@team02int1 www]# chmod -R 640 html/*
[root@team02int1 www]# ls -l html/
total 36
-rw-r-----. 1 root apache  149 Dec 11 00:27 about.html
-rw-r-----. 1 root apache 1372 Dec 12 12:20 commands.html
drw-r-----. 2 root apache   65 Dec 11 12:50 css
-rw-r-----. 1 root apache  766 Dec  5 10:23 favicon.ico
```

Website URL: http://10.134.178.2

Screenshots with the website

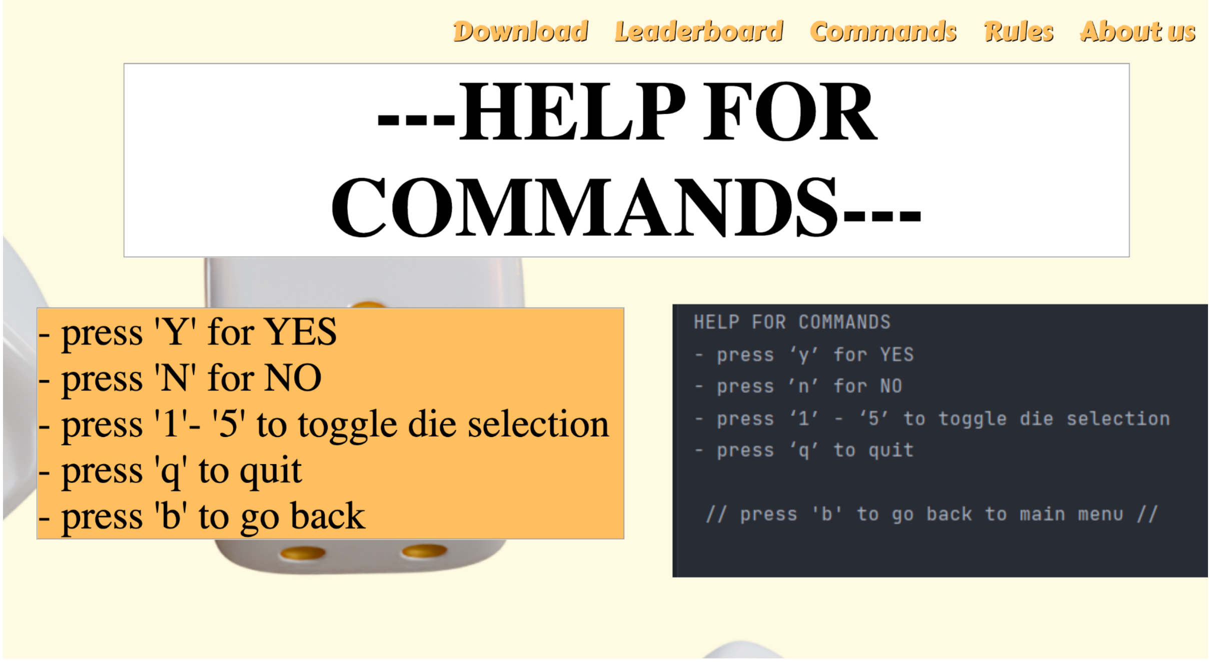






LEADERBOARD

Player1	430
Player2	580
Player3	310
Player4	640
Player5	495



---HELP FOR COMMANDS---

- press 'Y' for YES
- press 'N' for NO
- press '1'- '5' to toggle die selection
- press 'q' to quit
- press 'b' to go back

```
HELP FOR COMMANDS
- press 'y' for YES
- press 'n' for NO
- press '1' - '5' to toggle die selection
- press 'q' to quit

 // press 'b' to go back to main menu //
```

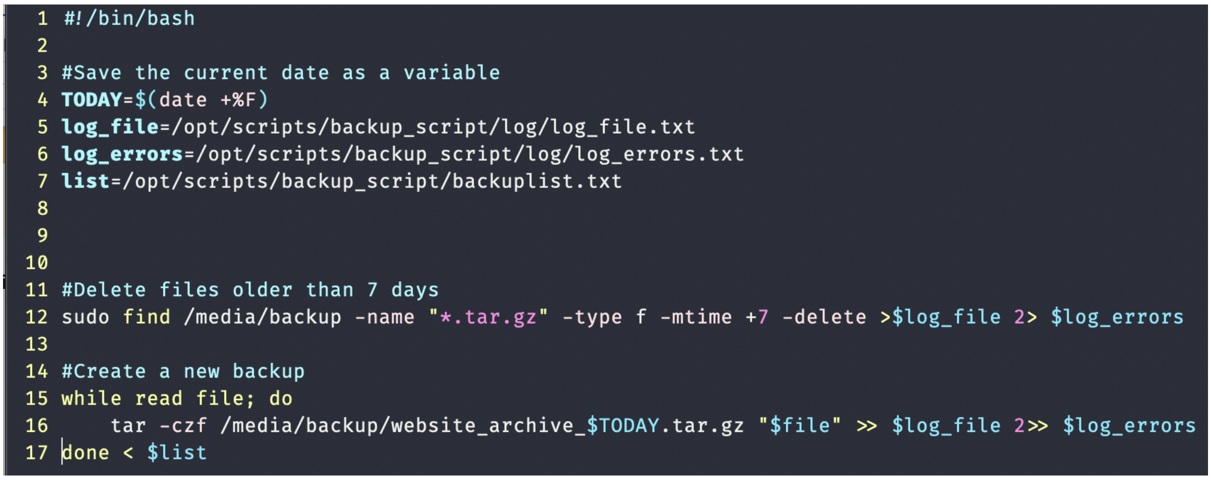
5. Create 2 of the proposed scripts

a. Backup Script

The purpose of the script is to create a backup of every file that is listed in the "/opt/backuplist" file. We are gonna use this script to create a daily backup of our website.

The script executes 2 commands. The first one deletes the file from the folder "/media/backup" that have the extension ".tar.gz" if they are older than 7 days. The output, if there is any, is saved in the "execution.log" file and the errors are sent to the "error.log" file. The second command reads the "backuplist" file and uses its content to create an gzip archive in the "/media/backup" folder.

The script will run everyday at 23:59, to modify this use the command "sudo crontab -e" and edit the first two columns



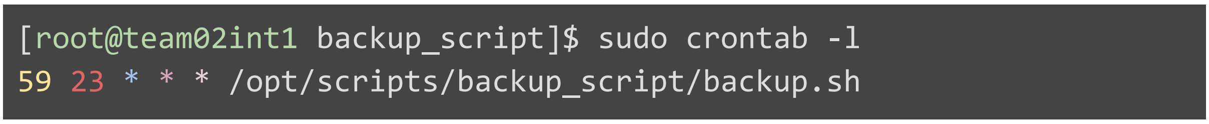
```
#!/bin/bash

#Save the current date as a variable
TODAY=$(date +%F)
log_file=/opt/scripts/backup_script/log/log_file.txt
log_errors=/opt/scripts/backup_script/log/log_errors.txt
list=/opt/scripts/backup_script/backuplist.txt



#Delete files older than 7 days
sudo find /media/backup -name "*.tar.gz" -type f -mtime +7 -delete >$log_file 2> $log_errors

#Create a new backup
while read file; do
    tar -czf /media/backup/website_archive_$TODAY.tar.gz "$file" >> $log_file 2>> $log_errors
done < $list
```



```
[root@team02int1 backup_script]$ sudo crontab -l
59 23 * * * /opt/scripts/backup_script/backup.sh
```

b. Create users script

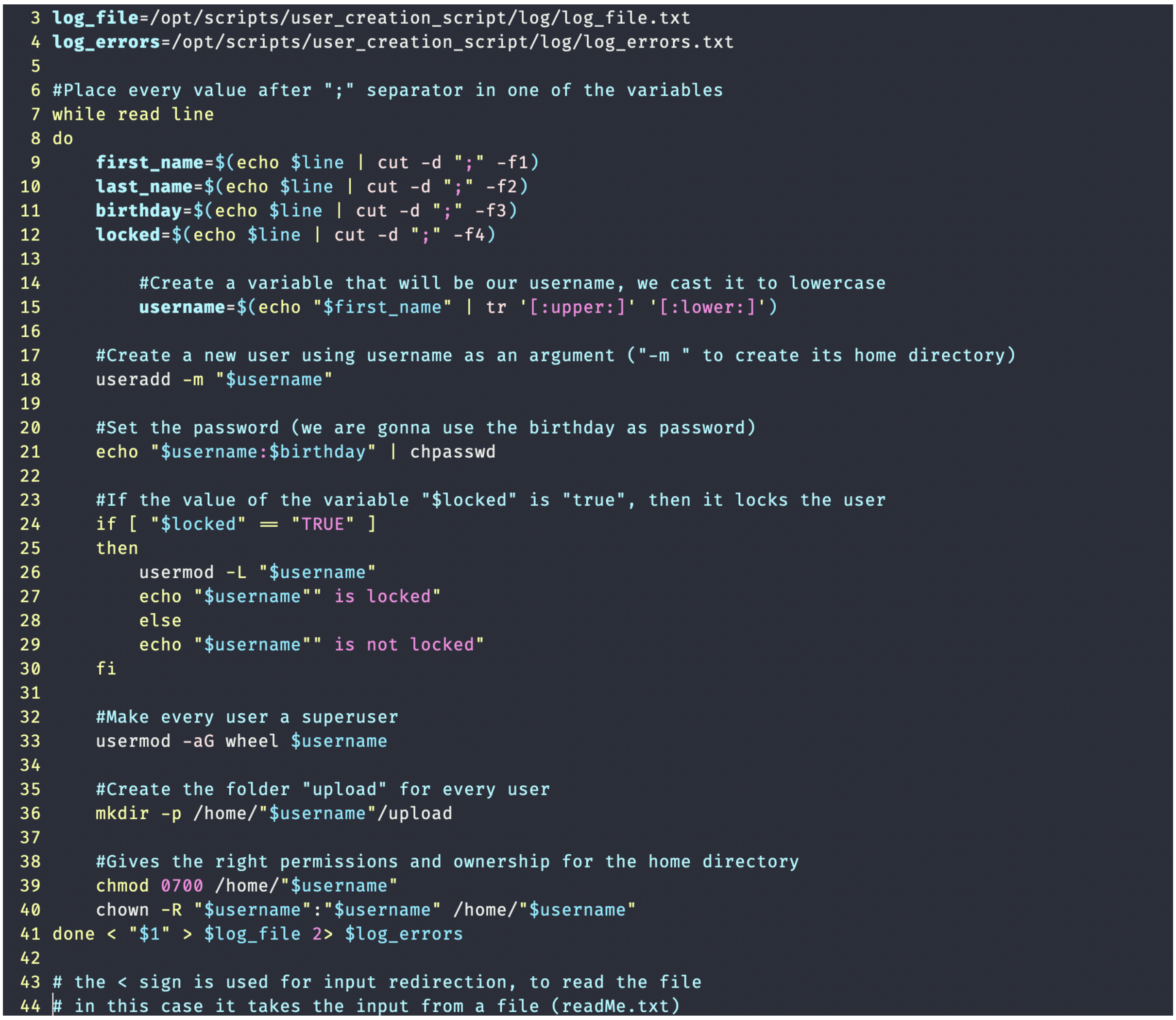
The purpose of this script is to create a new user for every record from a csv file

The file for the input must be used as an argument for the script
##Example##
./user_create.sh users_list.csv

The script uses the birthday as password for the new user, and locks it if the cvs specify it

The cvs file must have the following format
first_name;last_name;birthday;locked
Jayson;Krier;19.10.2001;FALSE
Nixie;Lutz;22.04.2000;TRUE

The errors are redirected to ./log/log_errors.txt
The output is redirected to ./log/log_file.txt



```
log_file=/opt/scripts/user_creation_script/log/log_file.txt
log_errors=/opt/scripts/user_creation_script/log/log_errors.txt

#Place every value after ";" separator in one of the variables
while read line
do
    first_name=$(echo $line | cut -d ";" -f1)
    last_name=$(echo $line | cut -d ";" -f2)
    birthday=$(echo $line | cut -d ";" -f3)
    locked=$(echo $line | cut -d ";" -f4)

        #Create a variable that will be our username, we cast it to lowercase
        username=$(echo "$first_name" | tr '[:upper:]' '[:lower:]')

    #Create a new user using username as an argument ("-m " to create its home directory)
    useradd -m "$username"

    #Set the password (we are gonna use the birthday as password)
    echo "$username:$birthday" | chpasswd

    #If the value of the variable "$locked" is "true", then it locks the user
    if [ "$locked" == "TRUE" ]
    then
        usermod -L "$username"
        echo "$username"" is locked"
        else
        echo "$username"" is not locked"
    fi

    #Make every user a superuser
    usermod -aG wheel $username

    #Create the folder "upload" for every user
    mkdir -p /home/"$username"/upload

    #Gives the right permissions and ownership for the home directory
    chmod 0700 /home/"$username"
    chown -R "$username":"$username" /home/"$username"
done < "$1" > $log_file 2> $log_errors

# the < sign is used for input redirection, to read the file
# in this case it takes the input from a file (readMe.txt)
```