1. Dayyana Dimitrova
2. 1000106454
3. Last updated -9/12/2022

Task 1- ScriptTask1.txt contains the bash script with notes on creating the tasks from task 1.

```
  GNU nano 6.2                          assessdayy.sh
#!/bin/bash
file=$1
IFS=";"

amount=$(wc -l < $file)
echo "You will add $amount users from $1, proceed?"
read -p "(yes/no) " answer
echo
if [[ $answer == "Yes" || $answer == "yes" ]]; then

echo >> userLog.txt
echo "Add users">> userLog.txt
(TZ="NZ" date) >> userLog.txt #add current time stamp to userLog

while read email dob groupys folder
do

echo "$email    $dob  $groupys $folder"
#create username
lastName=$(echo $email | awk -F '[@.]' '{print $2}')
username=${email:0:1}$lastName
#create password
password=$(echo $dob |awk -F '[/;]' '{print $2.$1}')

#create first group (if not exist)
```

```
  GNU nano 6.2                          assessdayy.sh
#create first group (if not exist)
group=$(echo $groupys | awk -F '[.]' '{print $1}')
grep -q "\b$fgroup\b" /etc/group>> /dev/null
if [ $? -eq 0 ];then
echo >> /dev/null
else
echo >> /dev/null
sudo groupadd -f "$fgroup"
fi

#create secondary group (if not exist)
sgroup=$(echo $groupys | awk -F '[,]' '{print $2}')
grep -q "\b$sgroup\b" /etc/group >> /dev/null
if [ $? -eq 0 ] ;then
echo >> /dev/null
else
echo >> /dev/null
sudo groupadd -f $sgroup
fi

#creare users
if [ -z "$groupys" ]
then
useradd -d /home/$username -m -s /bin/bash $username
else
```

```
else
useradd -d /home/$username -m -s /bin/bash -G $groupys $username
fi

#create password for user and force change on first login
echo $username:$password | chpasswd
chage --lastday 0 $username

#separate groups
Groups=$(echo $groupys | awk -F '[,]' '{print $1,$2}')

echo "Added user $username, with the password $password, in group(s) $Groups"
echo "Added user $username, with the password $password, group(s) $Groups" >> 

#create sharefolder group
if [ -z "$folder" ]
then
echo "Not a part of folder group" >> userLog.txt
else
#creating specified name for shareGroup dependent in folder
share=$(echo $folder | awk -F '[/]' '{print $2}')
shared=${share::-4}
shareGroup=${share::-3}
sudo groupadd -f $shareGroup
sudo usermod -a -G $shareGroup $username
```

```
sudo usermod -a -G $shareGroup $username
echo "Folder group is $shareGroup">> userLog.txt
fi

#Create shared folder
if [ -d "$folder" ] || [ -z "$folder" ]
then
echo >> /dev/null
else
echo >> /dev/null
sudo mkdir -m 770 $folder
sudo chgrp -R $shareGroup $folder
fi

#create symbolic link for sudo users
if [ -z "$folder" ]
then
echo "no shared folder" >> userLog.txt
else
sudo ln -s $folder /home/$username/shared
fi

#create alias for sudo
if getent group sudo | grep -q "\b${username}\b";
then
```

```
#create alias for sudo
if getent group sudo | grep -q "\b${username}\b";
then
echo "User is part of sudo" >> userLog.txt
touch /home/$username/.bash_aliases
sudo chown $username /home/$username/.bash_aliases
sudo chmod 700 /home/$username/.bash_aliases

echo "alias myls='ls -lisa'" >> /home/$username/.bash_aliases
source /home/$username/.bash_aliases
else
echo "user is not part of sudo" >> userLog.txt
fi

echo
echo >> userLog.txt

#finish reading the file
done < $file

else
echo ""
echo "Users not added"
fi
```

Outcome:

```
You will add 9 users from user.csv, proceed?
(yes/no) yes

edsger.dijkstra@tue.nl    1930/05/11  sudo,staff /staffData
useradd: user 'edijkstra' already exists
Added user edijkstra, with the password 051930, in group(s) sudo staff

john.mccarthy@caltech.edu    1927/09/04  sudo,visitor /visitorData
Added user jmccarthy, with the password 091927, in group(s) sudo visitor

andrew.tanenbaum@vua.nl    1944/03/16  staff /staffData
Added user atanenbaum, with the password 031944, in group(s) staff

alan.turing@cam.ac.uk    1912/06/23  visitor /visitorData
Added user aturing, with the password 061912, in group(s) visitor

linus.torvalds@linux.org    1969/12/28  sudo
Added user ltorvalds, with the password 121969, in group(s) sudo
```

Task 2- With the back-up script (Backup.txt file) we create a server (since it was not given to us by assignment sheet), we choose a remote directory and a port number. We set the conditions for the directory with the use of exits as well.

Then we create the backup process starting stage and make sure to user tar to compress our files.

```
  GNU nano 6.2                          backupdayy.sh *
#!/bin/bash
# Set variable for remote server information
remoteserv="enter a server"
remotedir="enter directory"
portnumb="enter port number"
if [ $# -eq 1 ]; then
if [ ! -d $1 ]; then
echo "Directory not found:$1"
echo "Exit"
exit 1
fi
else
exit 1;
fi
sourceDir=$1
targzFile=${sourceDir##*/}.tar.gz
echo "Backup Directory:${sourceDir}"
echo "compressed filename:${targzFile}"
echo "Remote Server:${remoteserv}"
echo "Server Directory:${remotedir}"
#-------------------------------------
# Backup process start

echo "-----------------------------------"
echo "Backup process starts"
```

```
targzFile=${sourceDir##*/}.tar.gz
echo "Backup Directory:${sourceDir}"
echo "compressed filename:${targzFile}"
echo "Remote Server:${remoteserv}"
echo "Server Directory:${remotedir}"
#-------------------------------------
# Backup process start

echo "-----------------------------------"
echo "Backup process starts"
echo ">> Compressing files"
#sudo tar czf ${targzFile} ${sourceDir} 2> /dev/null >&1
sudo tar czf ${targzFile} ${sourceDir} 2> /dev/null >&1
echo "-----------------------------------"
echo ">> Copy backup file to remote server"
#sudo scp -P 22 ${targzFile} ${remoteserv} "${remotedir}" 2> /dev/null >&1 # F>
sudo scp -P ${portnumb} ${targzFile} ${remoteserv}:"${remotedir}" 2> /dev/null>
#sudo scp -P ${portnumb} ${targzFile} ${remoteserv}:"${remotedir}"
if [[ $? -ne 0 ]]; then
echo "Backup failed"
else
echo "Backup completed"
fi
```