

Combine all descriptions into a single README

```
readme_content = f"""\
```

Linux Admin Automation Projects

This repository contains two Bash scripts created for CYB-300 (System and Communication Security) focused on automating core administrative tasks in Linux environments.

```
#!/bin/bash #Marjean Mayo-Baker #4/15/2025 #Cyb-300-14656-M01 -system and comm
security
```

define variables

#place the name of the groups in an array without defining the index,becuae we are not accessing the indexing when accessing the variable, but using a for do loop that iterates through to create a group in the /etc/ groupName=("Human_Resources" "Finance" "Sales")

itereate through the three groups to define if the groups exist if not to create it and then print which is true first this is funcation of a conditional statment

```
for i in "${groupName[@]}"; do
```

if the group matching i is true when the code iterates it will print already exists.

```
if getent group "$i" ; then
echo "Group already exists."
```



else

**if the group doesn't match the iteration (i)
create and print created successfully**

```
sudo groupadd "$i"  
echo "group '$i' created successfully."
```



exit out of loop

fi done

**define another loop, decided to do a
seperate loop because nested loops confuse
me and i never can get them to work. Also
it's easier to look at the script and edit
seperate.**

#using j as the new variable iterator not confuse previous loop

this loops through 1-12 to create 12 groups

```
for j in {1..12}; do
```

**variable that holds the users and the iterator
j to print out and keep track**

```
username="Users$j"
```

if the id matches existing on (error control)

```
if id "$username"; then
echo "User '$username' already exists."
else
```



else username isn't found set user name and print

```
sudo useradd "$username"
echo "${username}:NewP@ssw0rd" | sudo chpasswd
```



```
#and add the users to the sales group sudo usermod -a -G Sales "$username" echo
"Created user '$username' with default password 'NewP@ssw0rd' and added 'Sales'
group." fi donenAdd_cleaned.sh...]()
```

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

this is to create the back up folder

```
mkdir -p "$HOME/Desktop/user1" mkdir -p "$HOME/Desktop/user2" #Marjean Mayo-Baker
#CYB-300-14656-M01 system Comm security
```

define the variable

```
#i am going to create two backup location for redundancy
folders=("$HOME/Desktop/user1"
"$HOME/Desktop/user2") #iteration in the folders for do
loop that has a conditional satement for error checking for
dir in "${folders[@]}"; do name=$(basename "$dir")
backupFile="$HOME/Desktop//${name}_backup.tar.gz" #if
the folder backup file already exists print already exist if [ -
f "$backupFile" ]; then echo "Back up $name already
```

exisits" else sudo tar -czvf "\$backupFile" "\$dir" echo
"Backup for *namecreatedsuccessfully*. "*fidoneecho*"(ls -l)"

```
1 ip a
2 sudo -i
3 sudo apt full-upgrade
4 sudo -i
5 exit
6 sudo -i
7 exit
8 ip a
9 ip a
```



```
10 passwd 11 touch adminAdd.sh 12 nano adminAdd.sh 13 chmod +x adminAdd.sh 14
./adminAdd.sh 15 nano adminAdd.sh 16 ./adminAdd.sh 17 nano adminAdd.sh 18
./adminAdd.sh 19 nano adminAdd.sh 20 ./adminAdd.sh 21 nano adminAdd.sh 22 clear 23
nano backup 24 pwd 25 touch user1 26 thouch user2 27 touch user2 28 ls -l 29 nano
backup.sh 30 la 31 nano backup.sh 32 chomd +x backup.sh 33 chmod +x backup.sh 34
./backup.sh 35 nano backup.sh 36 ./backup.sh 37 nano backup.sh 38 ./backup.sh 39 nano
backup.sh 40 ./backup.sh 41 nano backup.sh 42 ./backup.sh 43 nano backup.sh 44
./backup.sh 45 nano backup.sh 46 ./backup.sh 47 ls 48 nano backup.sh 49 ./backup.sh 50
nano backup.sh 51 ./backup.sh 52 nano backup.sh 53 ./backup.sh 54 nano backup.sh 55
./backup.sh 56 history > ~/Desktop/terminal_history ading terminal_history...]()
```

Author

Marjean Mayo-Baker