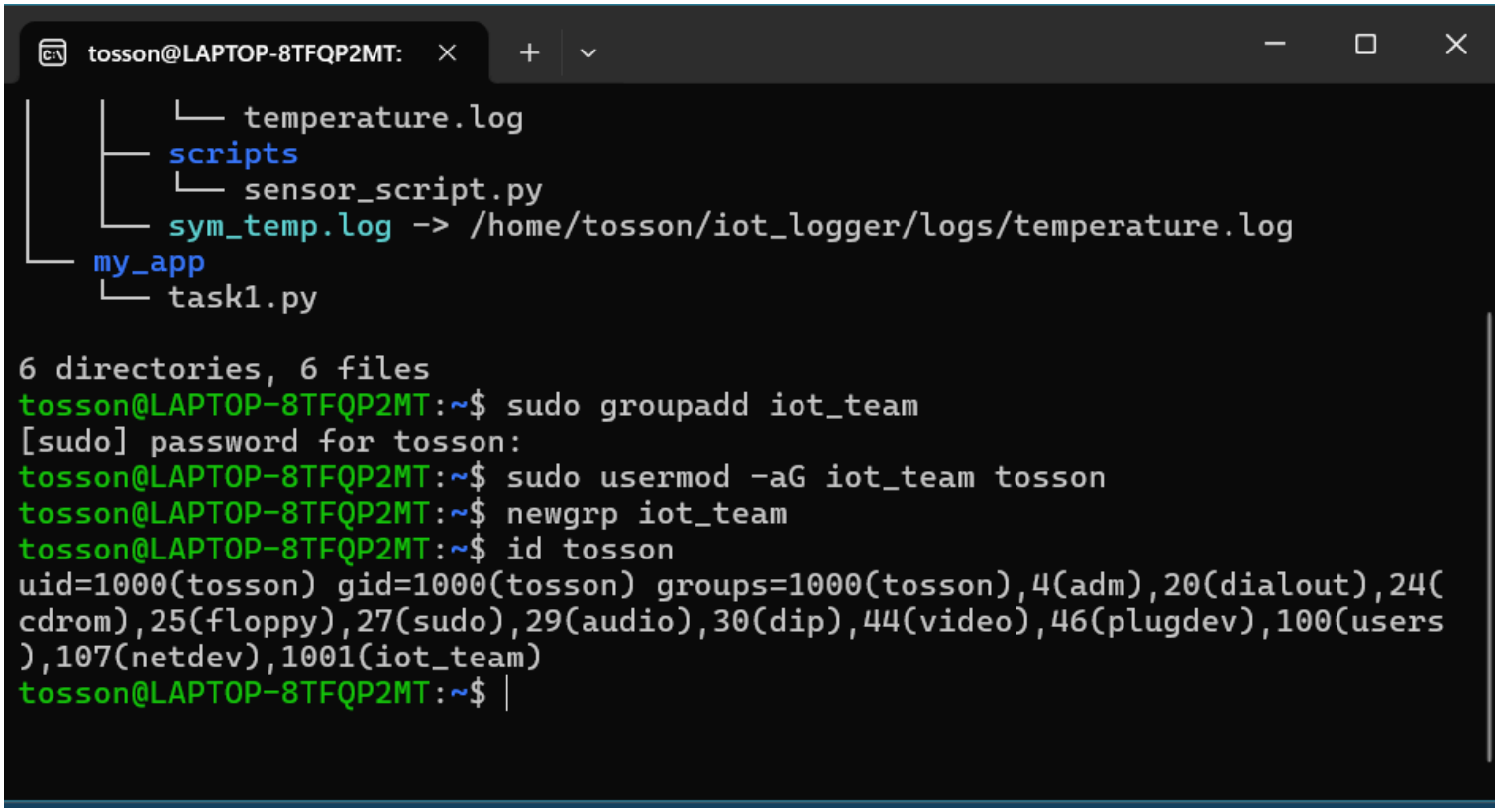
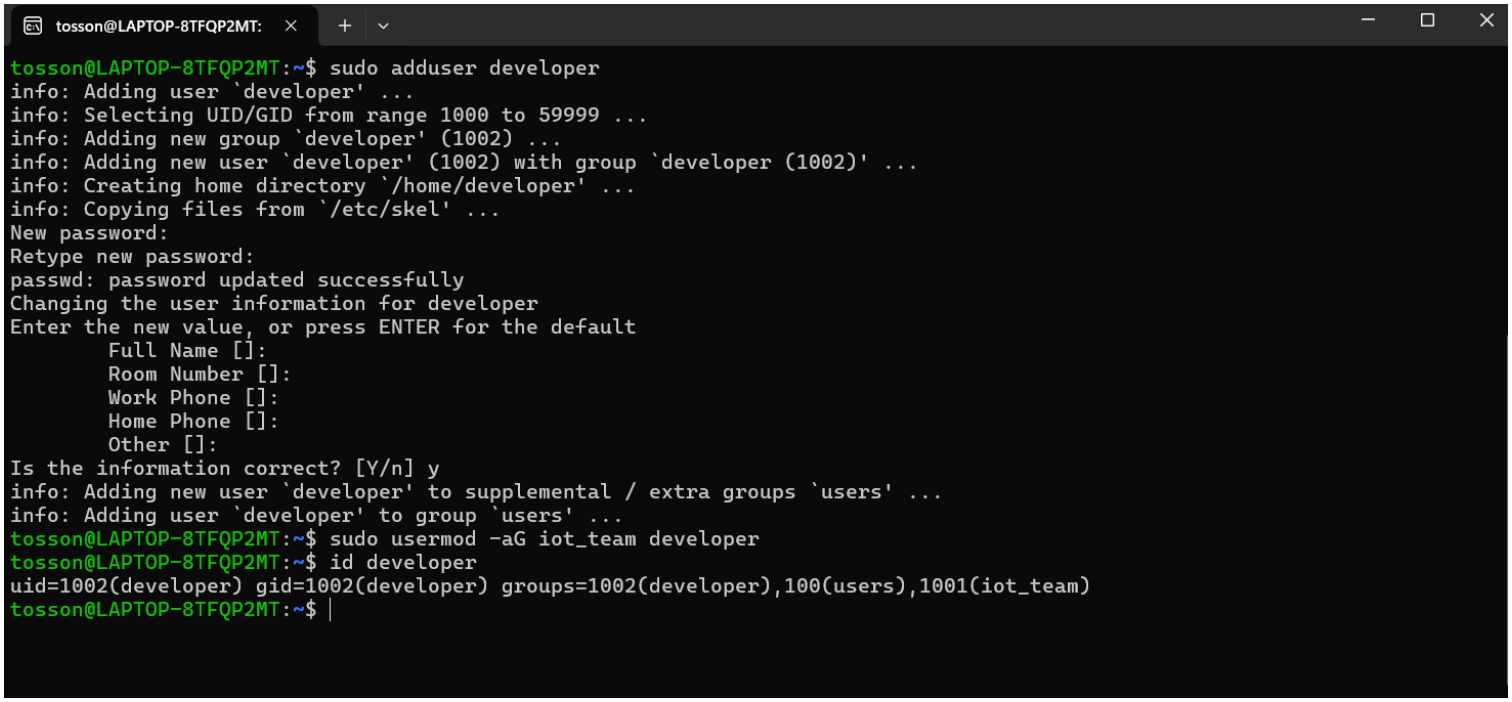
Day3

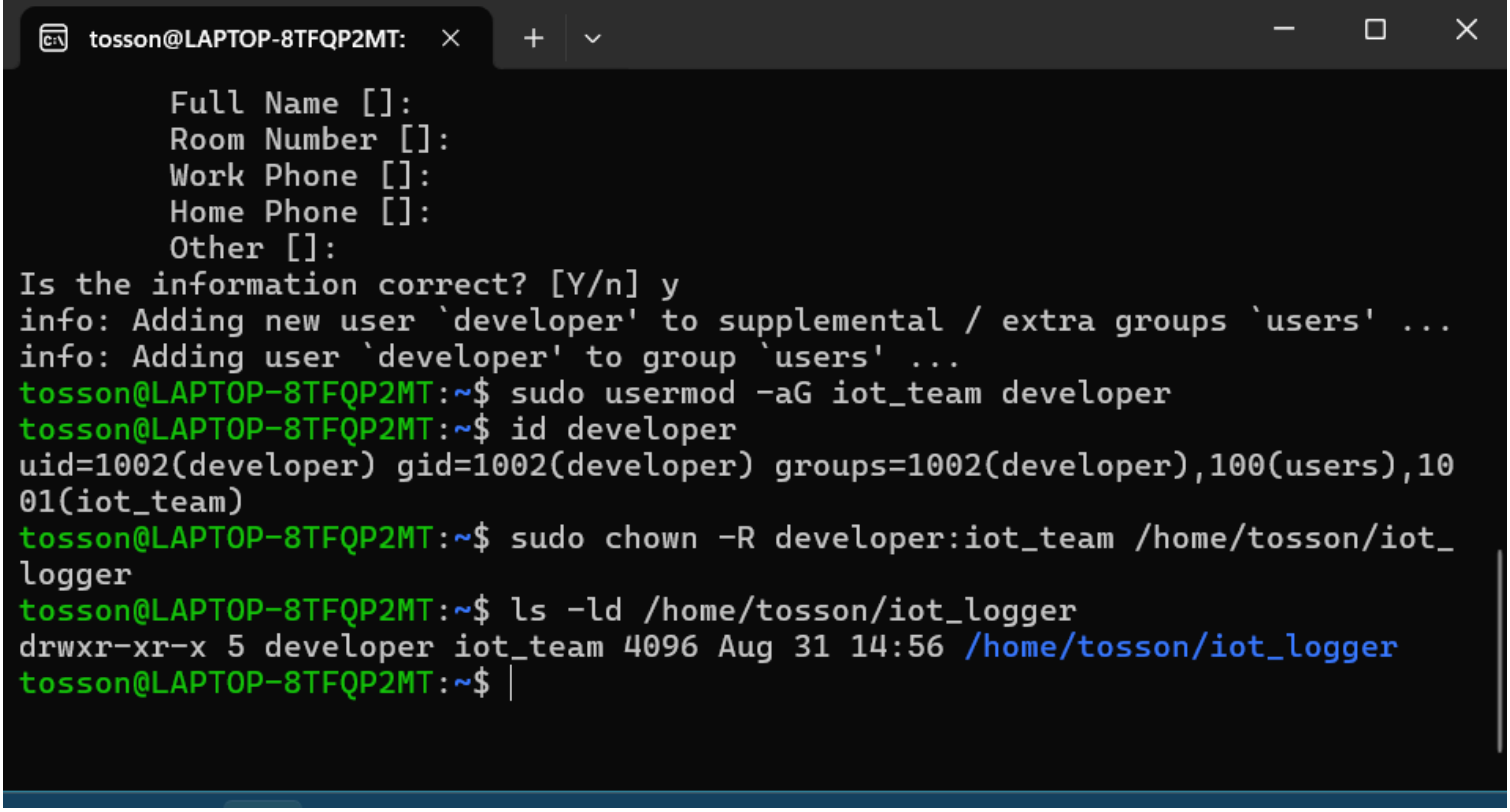
Create a new group iot_team and add your user to it.



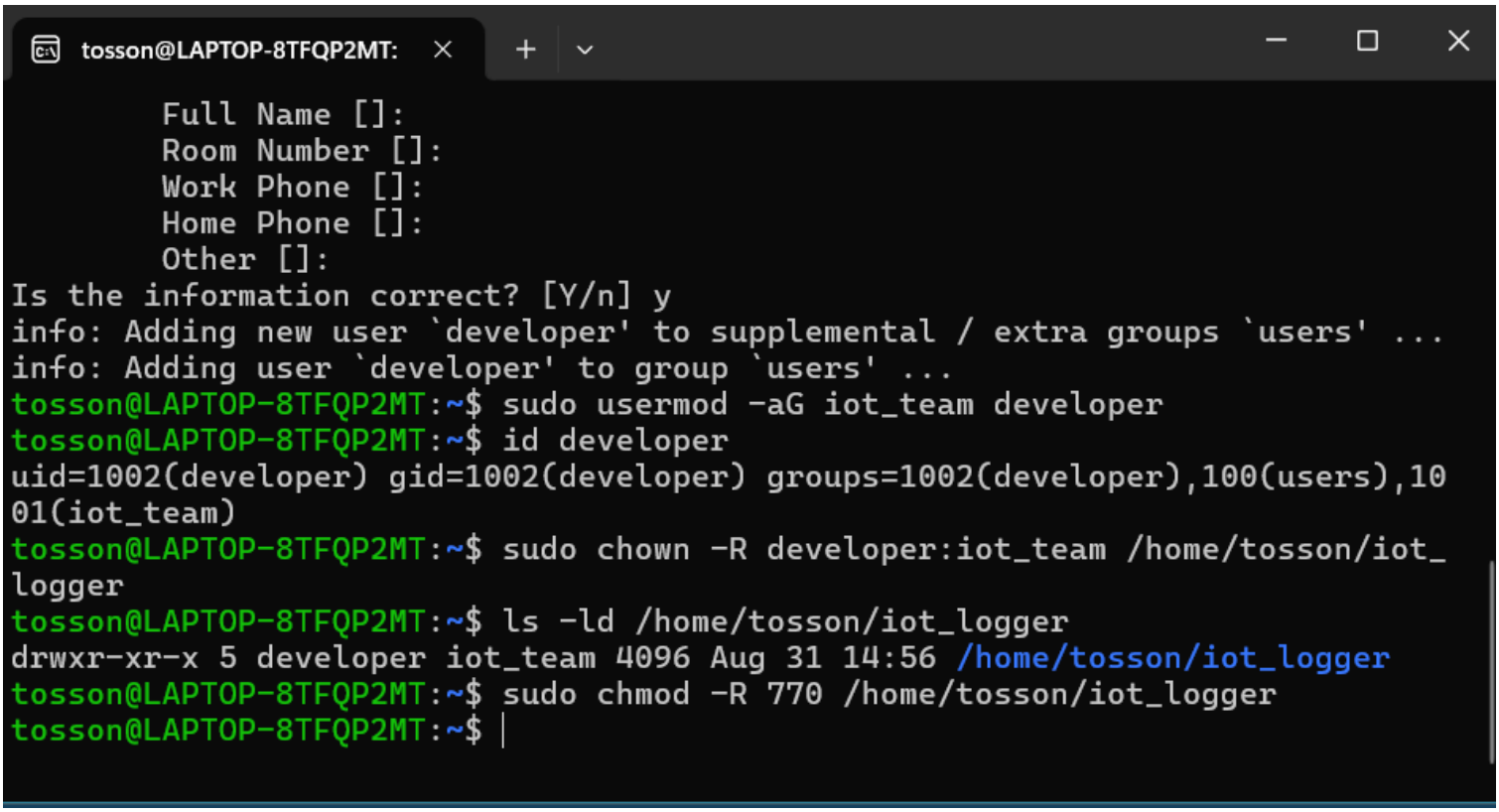
Create a new developer user, add it to the group.



Change ownership of iot_logger to the developer + group.



Set permissions: group can read/write logs, others blocked.



Test access as new user, then remove test user.

```
tosson@LAPTOP-8TFQP2MT:
developer@LAPTOP-8TFQP2MT:~$ exit
logout
tosson@LAPTOP-8TFQP2MT:~$ sudo chmod o+x /home/tosson
[sudo] password for tosson:
tosson@LAPTOP-8TFQP2MT:~$ ls -ld /home/tosson/
drwxr-x--x 8 tosson tosson 4096 Aug 31 13:35 /home/tosson/
tosson@LAPTOP-8TFQP2MT:~$ su - developer
Password:
developer@LAPTOP-8TFQP2MT:~$ cd /home/tosson/iot_logger/logs
developer@LAPTOP-8TFQP2MT:/home/tosson/iot_logger/logs$ echo "test line from developer" >> tempe
rature.log
developer@LAPTOP-8TFQP2MT:/home/tosson/iot_logger/logs$ cat temperature.log
test line from developer
developer@LAPTOP-8TFQP2MT:/home/tosson/iot_logger/logs$ exit
logout
tosson@LAPTOP-8TFQP2MT:~$ sudo deluser --remove-home developer
info: Looking for files to backup/remove ...
info: Removing files ...
info: Removing crontab ...
info: Removing user `developer' ...
tosson@LAPTOP-8TFQP2MT:~$ id developer
id: ‘developer’: no such user
tosson@LAPTOP-8TFQP2MT:~$
```

How do Linux file permissions (r, w, x) work for files vs directories? Give an example using ls -l.

in directories: (r) list the names of files inside
(w) create or delete files
(x) lets you enter the directory with (cd) and access its contents

in files: (r) you can read the content
(w) allows you to modify or delete the file
(x) lets you execute it as a program or script

example:

```
tosson@LAPTOP-8TFQP2MT:~$ ls -l
total 8
drwxrwx--- 5   1002 iot_team 4096 Aug 31 14:56 iot_logger
drwxr-xr-x 2 tosson tosson   4096 Aug 28 11:36 my_app
tosson@LAPTOP-8TFQP2MT:~$
```

Explain octal notation for permissions and what the umask command does. Give one calculation example.

each permission set (owner, group, others) is given a number: `r=4`, `w=2`, `x=1`. By adding them, you get a single digit for each set. For example, `rwx` = 7, `rw-` = 6, `r--` = 4, so a file with `rw-r--r--` equals 644
The `umask` command defines the default permissions for new files and directories by “masking out” certain bits. The system usually starts with defaults of `666` for files and `777` for directories, then subtracts the `umask` value.
example :

if the umask is `022`, then new files get `666 - 022 = 644` (`rw-r--r--`), and new directories get `777 - 022 = 755` (`rwxr-xr-x`).

What is the difference between the root user and a normal user? Why is root considered dangerous?

The root user in Linux has full control over the system, while a normal user is limited to their own files and basic tasks. Root is considered dangerous because one wrong command (like deleting a system folder) can damage the whole OS or cause security risks, so it’s safer to use normal accounts and only switch to root when needed.