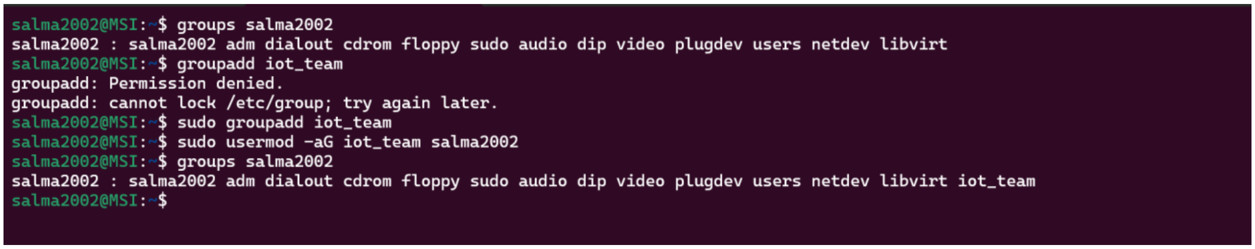
Day 3 – Phase 3: User, Group, and Permissions Management

Boss's Request: Secure the project and restrict access to authorized users only.

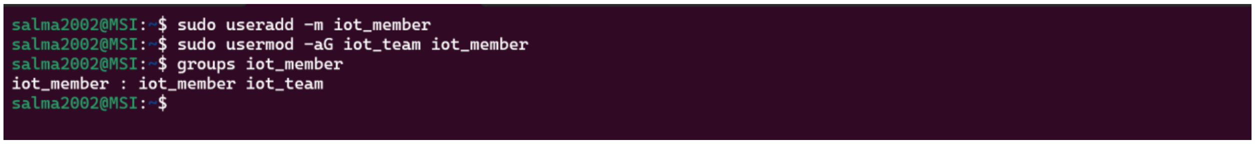
Tasks:

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



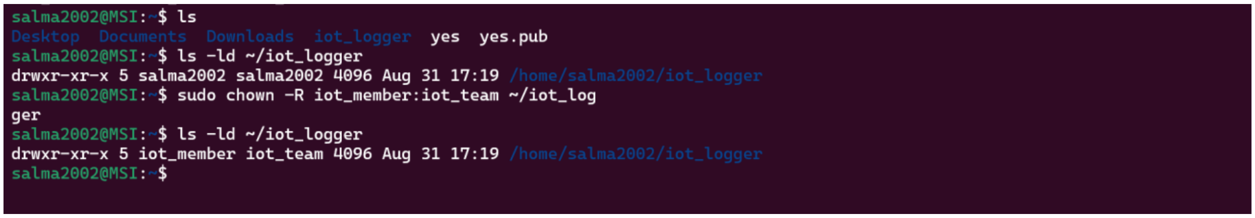
```
salma2002@MSI:~$ groups salma2002
salma2002 : salma2002 adm dialout cdrom floppy sudo audio dip video plugdev users netdev libvirt
salma2002@MSI:~$ groupadd iot_team
groupadd: Permission denied.
groupadd: cannot lock /etc/group; try again later.
salma2002@MSI:~$ sudo groupadd iot_team
salma2002@MSI:~$ sudo usermod -aG iot_team salma2002
salma2002@MSI:~$ groups salma2002
salma2002 : salma2002 adm dialout cdrom floppy sudo audio dip video plugdev users netdev libvirt iot_team
salma2002@MSI:~$
```

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



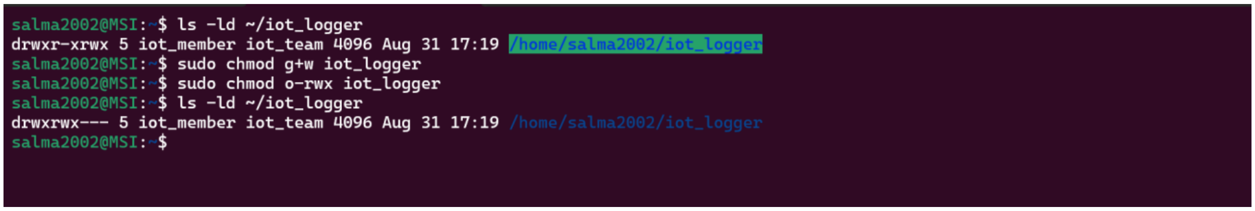
```
salma2002@MSI:~$ sudo useradd -m iot_member
salma2002@MSI:~$ sudo usermod -aG iot_team iot_member
salma2002@MSI:~$ groups iot_member
iot_member : iot_member iot_team
salma2002@MSI:~$
```

- Change ownership of iot_logger to the developer + group.



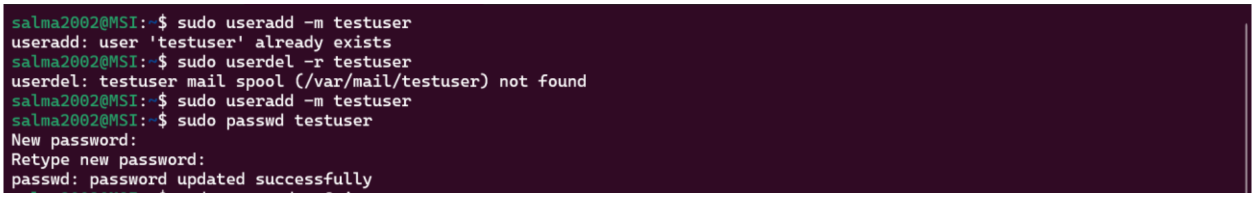
```
salma2002@MSI:~$ ls
Desktop  Documents  Downloads  iot_logger  yes  yes.pub
salma2002@MSI:~$ ls -ld ~/iot_logger
drwxr-xr-x 5 salma2002 salma2002 4096 Aug 31 17:19 /home/salma2002/iot_logger
salma2002@MSI:~$ sudo chown -R iot_member:iot_team ~/iot_log
ger
salma2002@MSI:~$ ls -ld ~/iot_logger
drwxr-xr-x 5 iot_member iot_team 4096 Aug 31 17:19 /home/salma2002/iot_logger
salma2002@MSI:~$
```

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

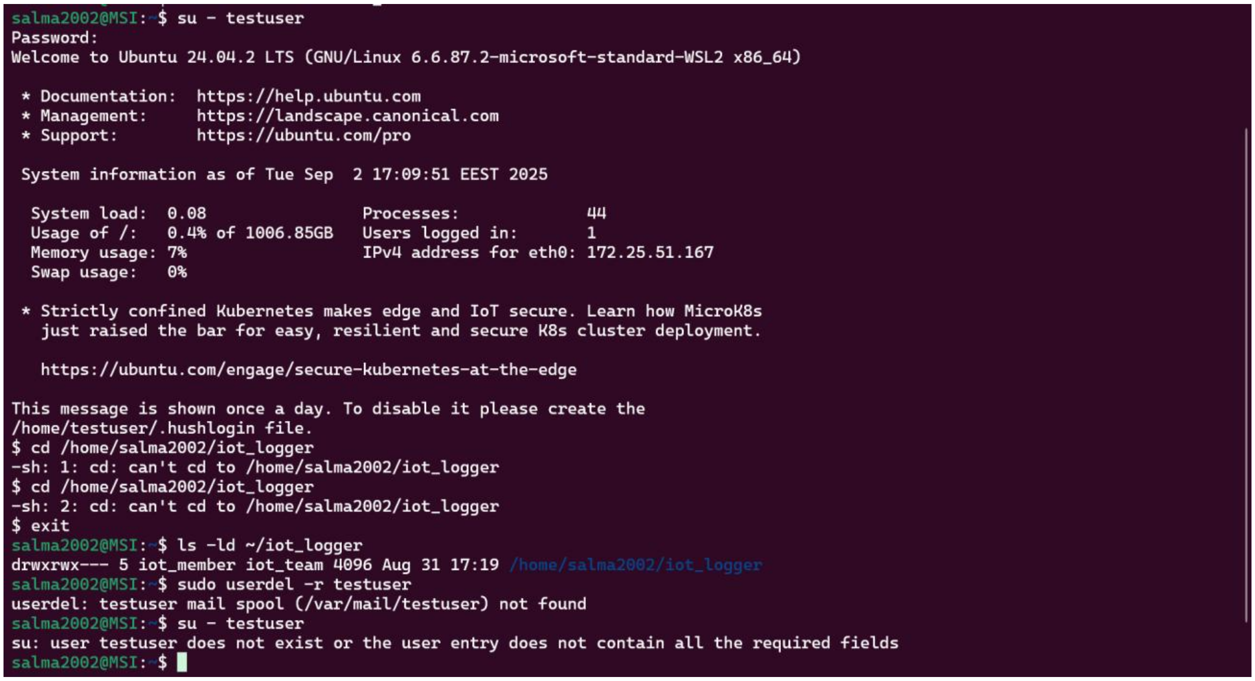


```
salma2002@MSI:~$ ls -ld ~/iot_logger
drwxr-xrwx 5 iot_member iot_team 4096 Aug 31 17:19 /home/salma2002/iot_logger
salma2002@MSI:~$ sudo chmod g+w iot_logger
salma2002@MSI:~$ sudo chmod o-rwx iot_logger
salma2002@MSI:~$ ls -ld ~/iot_logger
drwxrwx--- 5 iot_member iot_team 4096 Aug 31 17:19 /home/salma2002/iot_logger
salma2002@MSI:~$
```

- Test access as new user, then remove test user.



```
salma2002@MSI:~$ sudo useradd -m testuser
useradd: user 'testuser' already exists
salma2002@MSI:~$ sudo userdel -r testuser
userdel: testuser mail spool (/var/mail/testuser) not found
salma2002@MSI:~$ sudo useradd -m testuser
salma2002@MSI:~$ sudo passwd testuser
New password:
Retype new password:
passwd: password updated successfully
```



```
salma2002@MSI:~$ su - testuser
Password:
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.6.87.2-microsoft-standard-WSL2 x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Tue Sep  2 17:09:51 EEST 2025

  System load:  0.08               Processes:             44
  Usage of /:   0.4% of 1006.85GB   Users logged in:       1
  Memory usage: 7%                 IPv4 address for eth0: 172.25.51.167
  Swap usage:   0%

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

This message is shown once a day. To disable it please create the
/home/testuser/.hushlogin file.
$ cd /home/salma2002/iot_logger
-sh: 1: cd: can't cd to /home/salma2002/iot_logger
$ cd /home/salma2002/iot_logger
-sh: 2: cd: can't cd to /home/salma2002/iot_logger
$ exit
salma2002@MSI:~$ ls -ld ~/iot_logger
drwxrwx--- 5 iot_member iot_team 4096 Aug 31 17:19 /home/salma2002/iot_logger
salma2002@MSI:~$ sudo userdel -r testuser
userdel: testuser mail spool (/var/mail/testuser) not found
salma2002@MSI:~$ su - testuser
su: user testuser does not exist or the user entry does not contain all the required fields
salma2002@MSI:~$
```

Open-Ended Questions:

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

For Files :

r (read) : you can view the contents (e.g., cat file.txt).

w (write) : you can modify the file (edit, truncate, remove content).

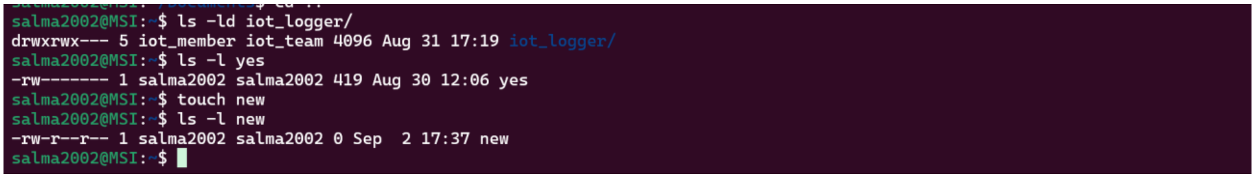
x (execute) : you can run the file as a program/script.

For Directories :

r (read) : you can list the files inside (ls).

w (write) : you can create, delete, or rename files inside (but not necessarily edit them).

x (execute) : you can enter the directory (cd) and access files *if you know their names*.



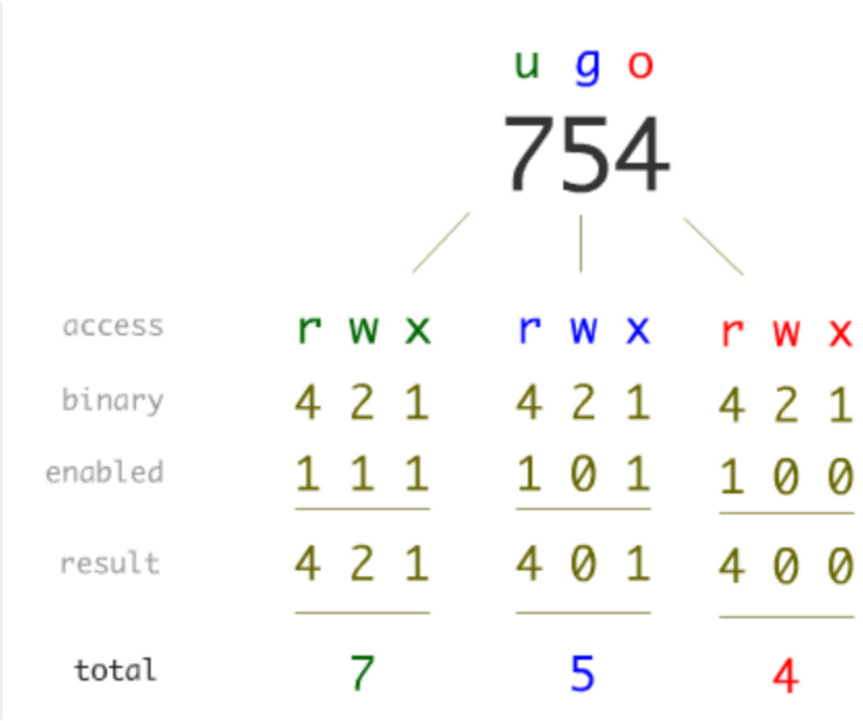
```
salma2002@MSI:~$ ls -ld iot_logger/
drwxrwx--- 5 iot_member iot_team 4096 Aug 31 17:19 iot_logger/
salma2002@MSI:~$ ls -l yes
-rw------- 1 salma2002 salma2002 419 Aug 30 12:06 yes
salma2002@MSI:~$ touch new
salma2002@MSI:~$ ls -l new
-rw-r--r-- 1 salma2002 salma2002 0 Sep  2 17:37 new
salma2002@MSI:~$
```

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

U stands for users

G stands for group

O stands for other



access	r	w	x	r	w	x	r	w	x
binary	4	2	1	4	2	1	4	2	1
enabled	1	1	1	1	0	1	1	0	0
result	4	2	1	4	0	1	4	0	0
total		7			5			4	

The umask command in Linux is used to set default permissions for files or directories the user creates.

- **File ->** The full permission set for a file is 666 (read/write permission for all)
- **Directory ->** The full permission set for a directory is 777 (read/write/execute)
- When we make a new directory, the permissions will be calculated as (full permissions for directory) - (umask value) i.e. 777 - 543 = 234
- When we make a new file, the permission will be given out similarly but with a slight change as follows: (full permissions for file) - (umask value) i.e. 666-543 = 123

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

Root User

- The **root** account is the **superuser** in Linux.
- It has **unrestricted access** to the entire system:
  - Can read, write, and execute any file (regardless of permissions).
  - Can install/remove software.
  - Can add/remove users or groups.
  - Can change ownership and permissions of any file.
  - Can shut down, reboot, or modify the kernel.

Normal User

- A normal (non-root) user has **limited privileges**:
  - Can only access files they own (or have permission for).
  - Cannot change system-critical files in directories like /etc, /bin, /usr.
  - Cannot install software system-wide (unless using sudo).
  - Can only manage processes they started.

The root user is dangerous because it can change and delete important files that can break the system