TRAINING DAY 12 REPORT:

- ## Working with Linux Files & Directories

  How to **create**, **read**, **write**, and **manage** files and folders. These are powerful tools I can use from the **terminal**.

1. Creating Files

Task	**Command Example**
Create empty file	touch file.txt
Create and edit	nano file.txt *(opens editor)*
Create with content	echo "Hello" > file.txt

2. Reading Files

Task	**Command Example**
View contents (short)	cat file.txt
View with scroll	less file.txt or more file.txt
Read only first lines	head -n 5 file.txt
Read only last lines	tail -n 10 file.txt

3. Writing to Files

Task	**Command Example**
Overwrite file	echo "new text" > file.txt
Append (add at end)	echo "add this" >> file.txt
Edit manually in terminal	nano file.txt

Tip: After editing in nano, press Ctrl + O to save, Ctrl + X to exit.

4. Creating Directories (Folders)

Task	Command Example
Create folder	mkdir myfolder
Create nested dirs	mkdir -p a/b/c

5. Deleting Files and Folders

Task	Command
Delete file	`rm file.txt`
Delete folder (empty)	`rmdir foldername`
Delete folder (with content)	`rm -r foldername`

6. Moving & Renaming

Task	Command
Move file	mv file.txt /path/folder/
Rename file	mv old.txt new.txt

7. Copying

Task	Command
Copy file	cp file.txt backup.txt
Copy folder	cp -r folder newfolder

- ## Adding a User to the Sudoers List

  How to **give admin (root) rights** to a user by adding them to the **sudoers list**.

What is the `sudoers` list?

- It’s a list of users allowed to run commands as root using `sudo`.

The config file is:

```
/etc/sudoers
```

Step-by-Step: Add a User to Sudoers

Step 1: Log in as root or a sudo user

```
su or sudo -i
```

Step 2: Add user to the `sudo` group

```
usermod -aG sudo username
```

Replace username with the actual username.

Step 3: (Optional) Verify

```
groups username
```

This will show if the user is in the `sudo` group.

Step 4: Test it

Login as the user and run:

```
sudo whoami
```

It should return:

```
root
```

- ## Working with Groups in Linux

How Linux uses **groups** to manage file access and organize users efficiently.

What is a Group in Linux?

A **group** is a collection of users.
It helps manage **permissions** for multiple users at once.

Every file belongs to:

- A **user (owner)**
- A **group**

1. Types of Groups

Group Type	**Description**
Primary	Main group assigned to a user
Secondary	Extra groups a user belongs to

Check a user's groups:

```
groups username
```

2. Common Group Commands

Task	**Command**
Create a group	`sudo groupadd groupname`
Delete a group	`sudo groupdel groupname`
Add user to group	`sudo usermod -aG groupname username`
Remove user from group	Remove manually from `/etc/group` (or use `gpasswd -d`)
See all groups	`cat /etc/group`
Change file's group	`sudo chgrp groupname filename`

3. Use Group Permissions on Files

Example: `ls -l file.txt`

Output: `-rw-rw-r-- 1 user **group** file.txt`

Here, the group **can read & write** the file.

To set group permission:

```
chmod g+w file.txt
```

Why Groups Are Useful

1. Share files among multiple users
2. Give project teams common access
3. Avoid changing ownership again and again

- ## APT Package Manager in Linux

How Linux installs, removes, and manages software using **APT (Advanced Package Tool)** — a command-line **package manager** for Debian-based systems like **Kali Linux**.

What is APT?

- APT stands for **Advanced Package Tool**

- Used to **install, update, upgrade, and remove software**

- Works with `.deb` packages (Debian format)

- Downloads from **online repositories** listed in:

```
/etc/apt/sources.list
```

Basic APT Commands:

Task	Command
Search for a package	`apt search packagename`

Task	Command
Install a package	`sudo apt install packagename`
Remove a package	`sudo apt remove packagename`
Remove including config	`sudo apt purge packagename`
Update package lists	`sudo apt update`
Upgrade installed packages	`sudo apt upgrade`
Full system upgrade	`sudo apt full-upgrade`
Clean cache	`sudo apt autoclean`
Remove unused dependencies	`sudo apt autoremove`
Show info about a package	`apt show packagename`

How APT Works :

- APT connects to software **repositories** online
- Downloads the required `.deb` files
- Automatically installs **dependencies** too
- Uses config files stored in `/etc/apt/` directory