✅ 1. `git version`

👉 **What it does:** Shows which version of Git is installed
✔️ You ran this to **check if Git is installed and working**.

✅ 2. `git init`

👉 **What it does:** Creates a new Git repository in your current folder
✔️ You used this to **start tracking your project with Git**.

You ran it **twice**:

- Once in your home folder (`~`) — not needed for your project.
- Once inside `my-project` — ✅ this is the correct one.

✅ 3. `mkdir my-project`

👉 **What it does:** Creates a new folder named `my-project`
✔️ You used this to **start a new project** in its own folder.

✅ 4. `cd my-project`

👉 **What it does:** Enters the `my-project` folder
✔️ Now you're working inside your project directory.

✅ 5. `touch README.md`

👉 **What it does:** Creates a blank file named `README.md`
✔️ This file is commonly used to **describe your project**.

✅ 6. `git add README.md`

👉 **What it does:** Adds the file to the **staging area**
✔️ This means you're **telling Git to track this file** for the next commit.

✅ 7. `git commit -m "initial version"`

👉 **What it does:** Saves (commits) your staged files with a message
✔️ You **recorded your first snapshot** of the project.

✅ 8. `git config --global user.name "bharath"`

👉 **What it does:** Sets your Git **username** globally (applies to all projects)
✔️ Needed for Git to know who is making the commits.

✅ 9. `git config --global user.email "m.l.bharathmurugan@gmail.com"`

👉 **What it does:** Sets your Git **email** globally
✔️ Used to **identify the author** of commits.

✅ 10. `git add .`

👉 **What it does:** Adds **all files** in the current folder to staging
✔️ Easier way to add everything at once before committing.

✅ 11. `git commit -m "initial version"` (again)

👉 You ran this again after adding another file (`md`)
✔️ This committed the **new file** you added after the first commit.

✅ 12. `git config --global color.ui "auto"`

👉 **What it does:** Enables color output for Git in the terminal
✔️ Makes Git commands **easier to read** by using color.

✅ 13. `git config --global core.editor "'C:\Program Files\Sublime Text 3\sublime_text.exe'"`

👉 **What it does:** Sets Sublime Text as your **default Git editor**
✔️ This is used when Git needs you to type a longer message (like for merge commits or rebase).

✅ 14. `git config --list`

👉 **What it does:** Shows **all Git settings** currently configured
✔️ Useful to **check if your name, email, editor, etc. are set correctly**.

What You Did	Why You Did It
Created a folder	To start a new project
Initialized Git	To track changes with Git
Added and committed files	To save changes in Git history
Configured name and email	To identify yourself in commits
Set color and editor	To improve your Git experience

🔹 `mkdir myProjectDir`

👉 **What it does:** Creates a new folder named `myProjectDir`
✔️ Used to start a new project.

🔹 `cd myProjectDir/`

👉 **What it does:** Enters (moves into) the `myProjectDir` folder
✔️ You work inside this folder for your project.

🔹 `git init`

👉 **What it does:** Starts a new Git repository in this folder
✔️ Tells Git to begin tracking changes here.

🔹 `ls -ls`, `ls -lrt`, `ls -la`

👉 **What it does:** Lists files in the current folder with details
✔️ Used to **see what files or folders exist**.

> 📝 `ls -la` shows hidden files like `.git`.

🔹 `git status`

👉 **What it does:** Shows the current status of your Git repo
✔️ Used to check if there are any files to commit, or if you've made changes.

🔹 `pwd`

👉 **What it does:** Shows the current path (directory you're in)
✔️ Useful to confirm where you are in your system.

🔹 `cd ..`

👉 **What it does:** Goes **up one level** in the folder structure
✔️ Used to return from `myProjectDir` back to your home folder.

🔹 `git init newProjectDir`

👉 **What it does:** Creates a **new folder** named `newProjectDir` and initializes it as a Git repo (all in one step)
✔️ A shortcut way to both create the folder **and** set it up for Git.

🔹 `cd newProjectDir/`

👉 **What it does:** Enters the new project folder
✔️ Ready to start working and adding files here.

🔹 `ls -la`

👉 **What it does:** Lists all files including hidden ones like `.git/`
✔️ Confirms that Git was successfully initialized.

Command	Purpose
`mkdir myProjectDir`	Create new folder
`cd myProjectDir`	Enter the folder
`git init`	Start Git tracking
`ls -lrt` / `ls -la`	See files (normal or hidden)
`git status`	Check Git status
`pwd`	Show current folder path
`cd ..`	Go back one level
`git init newProjectDir`	Create + initialize Git repo in one step
`cd newProjectDir`	Enter the new project folder

INITIAL STAGES FOR COMMAND

```
# 1. Make a new directory for your project
mkdir myNoteDir

# 2. Go inside it
cd myNoteDir

# 3. Initialize Git repository in this folder
git init

# 4. Create files INSIDE this directory
touch README.md
touch index.html   # if you want another file

# 5. Add those files to staging
git add .

# 6. Commit your changes
git commit -m "Initial version"
```