SLIDE 1:

1. git init: Initialises a new Git repository in your project directory.
2. git add: Adds a file or changes to the staging area.
3. git commit: Saves changes to the local repository with a message describing the changes.
4. git push: Uploads local repository content to a remote repository.
5. git pull: Downloads and merges content from a remote repository to the local repository.
6. git log: Shows commit history.
7. git reset: Undo or adjust commit history.
8. git checkout: Switch branches or restore files.

git merge: Combine changes from branches.

**STEP 1: Create a new project directory**

mkdir git-practice

cd git-practice

✅ Now you're inside the new project folder.

**STEP 2: Initialize Git repository**

git init

This creates a hidden .git folder. Now Git starts tracking this folder.

**STEP 3: Create a new file (Untracked at first)**

echo "# My Portfolio" > resume.md

**Check Git status:**

git status

**Git will say: resume.md is Untracked**

**Because Git sees a new file but it's not staged yet.**

**STEP 4: Track the file by adding it to the Staging Area**

git add resume.md

git status

**Now Git shows:**

* **Changes to be committed:**
* **new file: resume.md**

**File is now in the Staging Area**

**STEP 5: Commit the file to Local Repository**

git commit -m "Initial resume created"

**Now it's in the Local Repository with a permanent commit message.**

**STEP 6: View Commit History**

git log –oneline

**You’ll see a commit ID and message**

**STEP 7: Modify the file (Back to Working Directory)**

echo "## Skills" >> resume.md

git status

**Git will say:**

* **Changes not staged for commit:**

**Because the file is Modified but not added back to Staging**

**STEP 8: Check what changed (Git Diff)**

git diff resume.md

**This shows the difference between the Working Directory and the last committed version.**

**STEP 9: Stage and Commit the new change**

git add resume.md

git commit -m "Added Skills section"

**STEP 10: Create multiple files to visualize more**

echo "Education details" > education.md

echo "Project details" > projects.md

git status

**Git will say both files are Untracked**

**STEP 11: Track only one file**

git add education.md

git status

**Git shows:**

* **education.md in Staging Area**
* **projects.md still Untracked**

**STEP 12: Commit the staged file**

git commit -m "Added education section"

**You can now compare how different file states behave:**

* **Tracked & committed (resume.md, education.md)**
* **Untracked (projects.md)**

**IN VISUAL STUDIO CODE:**

**Step 1: Create a New Folder & Open in VS Code**

Create a folder named git-practice on your desktop.

Open VS Code.

Go to File → Open Folder → Select git-practice.

**Step 2: Open the Terminal in VS Code**

Terminal → New Terminal

This opens a terminal *inside VS Code*, similar to Git Bash.

**Step 3: Initialize Git**

git init

This initializes a Git repo. You’ll see the VS Code Source Control icon (left sidebar) become active.

**Step 4: Create a New File**

In the Explorer sidebar:

1. Click on New File → name it resume.md
2. Add this content:

# My Portfolio

You’ve now created a new **Untracked file**

**Step 5: Check Status in Terminal**

git status

You’ll see resume.md as **Untracked**.

**Step 6: Stage the File**

git add resume.md

git status

Now the file is **Staged**.

**Step 7: Commit the File**

git commit -m "Initial resume created"

First commit is done — file is now in Local Repository.

**Step 8: Modify the File**

Add this in resume.md:

## Skills

- Python

- SQL

Save the file (Ctrl+S)

git status

File is now **Modified** in Working Directory

**Step 9: View Differences**

git diff resume.md

This will show what's changed.

**Step 10: Stage and Commit the Change**

git add resume.md

git commit -m "Added Skills section"

**Step 11: Add More Files**

echo "Education details" > education.md

echo "Project details" > projects.md

git status

You’ll see:

* education.md and projects.md as **Untracked**

**Step 12: Track One, Ignore the Other**

git add education.md

git commit -m "Added education section"

Now:

* education.md → Tracked
* projects.md → Still Untracked

**Step 13: Use .gitignore to Exclude Files**

Create a .gitignore file:

Add this inside:

touch .gitignore

Check:

git status

**Now projects.md should no longer appear — Git is ignoring it!**

**Go to Command Palette (Ctrl+Shift+P)**

**Type: Git Graph: View Git Graph → Click it**

**✅ A visual timeline of all commits will appear.**