# Day 10

## *Learned about Git command Line*

**Setup:**

- git config --global user.name "<Your Name>": Sets your name for commits

- git config --global user.email "<Your Email>": Sets your email for commits

- git init: Initializes a new Git repository in the current directory

**Tracking Changes:**

- git status: Shows the status of your working directory (modified, staged, untracked files)

- git add <file>: Adds a specific file to the staging area for the next commit

- git add .: Adds all modified files in the working directory to the staging area

- git reset <file>: Unstages a file from the staging area (keeps changes in working directory)

- git diff: Shows the difference between the working directory and the index (unstaged changes)

- git diff --staged: Shows the difference between the index and the HEAD (staged changes)

**Commits:**

- git commit -m "<message>": Creates a new commit with a descriptive message

**Branching and Merging:**

- git branch: Lists all local branches

- git branch <branch-name>: Creates a new branch

- git checkout <branch-name>: Switches to a different branch

- git merge <branch-name>: Merges changes from another branch into the current branch

    o Use git merge --ff-only <branch-name> for a fast-forward merge (no merge commit)

**Undoing Changes:**

- git stash: Temporarily stores uncommitted changes in a stash

- git stash pop: Applies the most recent stash to the working directory

- git stash list: Shows a list of stashes

**Viewing History:**

- git log: Shows the commit history of the current branch

    o -n <number>: limit commits shown

    o --oneline --graph --decorate: compact view with commit graph and references

**Ignoring Files:**

- Create a .gitignore file in your project root to specify files or patterns to exclude from version control