Git and GitHub Notes

Why We Use Git

- To track changes in our code or files.
- To go back to any previous version of the project.
- To work safely by experimenting without breaking the whole project.
- To collaborate with others without overwriting each other's work.

What is GitHub?

GitHub is a cloud-based website that allows you to:

- Store your Git projects online.
- Share your code with others (friends, team, or the public).
- Collaborate with others on the same project.

Why We Use GitHub

- Back up our projects online.
- Share code easily with others.
- Work as a team using tools like Pull Requests and Issues.
- Host open-source projects.

Starting a Git Project

When starting any project, we need to create a Git repository. This is done using:

`git init`

Basic Git Commands

- `git init` – Initializes a Git repository.
- `git add <filename>` – Adds a file to be tracked by Git. To add multiple files, list them with spaces. To add all files: `git add .`

- `git status` – Shows the current status and changes in the project.
- `git commit -m "message"` – Takes a snapshot of the code with a descriptive message.
- `git log` – Shows the commit history including author, date, and message.
- `git checkout <branch-name>` – Switches to another branch or a previous commit.
- `git checkout -b <branch-name>` – Creates and switches to a new branch. Useful when working on new features or changes.
- `git push` – Uploads your commits to a remote repository like GitHub.

Local vs Remote Branches

Local Branch:

- Exists only on your device.
- Created using `git checkout -b <branch-name>`.

Remote Branch:

- Exists on GitHub or another remote repository.
- Visible using `git branch -r`.

More Git Commands

- `git revert <commit>` – Reverts a previous commit by creating a new one that undoes the changes. This is safe for shared projects.
- `git reset --hard <commit>` – Resets your repository to a previous commit and permanently deletes later commits.
- `git merge <branch>` – Merges another branch into your current branch. Typically used to merge feature branches into `main`.
- `git branch -m <new-name>` – Renames the current branch.

Conclusion

Now that Git basics are covered, you are ready to move on to GitHub and remote collaboration.