

Git & GitHub: A Simple Overview

What is Git?

Git is a **version control system** that helps you track changes in your code, collaborate with others, and manage different versions of your project.

What is GitHub?

GitHub is a **cloud-based platform** where you can store your Git repositories online, collaborate with others, and manage projects.

How to Work with Git & GitHub (Real Project Example: A Website Project)

Step 1: Set Up Git and Create a New Project

You want to build a website project and track changes using Git.

```
git init
```

Creates a Git repository in your project folder.

```
git remote add origin <your_github_repo_url>
```

Links your local project to a GitHub repository.

Step 2: Save Changes (Add & Commit)

You write some HTML and CSS files. Now, you want to save them.

```
git add .
```

Stages all new/modified files for commit.

```
git commit -m "Added homepage layout"
```

Saves changes with a message.

```
git push -u origin main
```

Uploads your project to GitHub.

Step 3: Make More Changes & Track Versions

You add a "Contact Us" page.

```
git add .  
git commit -m "Added contact page"  
git push
```

Every commit keeps a history of changes, so you can always go back!

Step 4: Work on a New Feature Using Branches

You want to add a "Dark Mode" feature but don't want to break the main code.

```
git checkout -b dark-mode
```

Creates a new branch for the feature.

You add the dark mode code, then:

```
git add .  
git commit -m "Added dark mode feature"  
git push -u origin dark-mode
```

Step 5: Merge Changes Back to Main Code

Once the dark mode is ready, you merge it into the main project.

```
git checkout main  
git merge dark-mode  
git push
```

Merges the feature into the main branch.

You can now delete the feature branch:

```
git branch -d dark-mode
```

Step 6: Collaborate with Others

- Your team can pull the latest code:

```
git pull origin main
```

- They can create new branches for their work.
 - You can review & merge their changes via **Pull Requests** on GitHub.
-

Step 7: Release a Version

Once your project is ready, tag it as a **release version**:

```
git tag -a v1.0 -m "First official release"  
git push origin v1.0
```