Git and GitHub

What is Git?

Git is a **version control system (VCS)** that helps developers manage changes in their code over time. It allows developers to:

- ☑ Track changes in a project.
- ☑ Collaborate with multiple developers.
- ☑ Work on different features using branches.
- ☑ Roll back to previous versions if needed.

What is GitHub?

GitHub is an **online platform** for hosting Git repositories. It provides:

- ☑ Cloud storage for Git projects.
- ☑ Collaboration tools for teams.

☑ A web interface to manage repositories.

☑ Security and backup for code.

GitHub allows developers to **store, share, and contribute** to projects efficiently.

Installing Git

Before using Git, it needs to be installed on your computer.

Installing Git on Windows

1. Download Git from https://git-scm.com/
2. Run the installer and follow the setup instructions.
3. Open **Git Bash** to start using Git.

Installing Git on Mac

Use Homebrew to install Git:

```
brew install git
```

Installing Git on Linux

For Ubuntu/Debian:

```
sudo apt install git
```

Basic Git Commands

Command	Description
`git --version`	Check Git version installed.
`git config --global user.name "Your Name"`	Set your Git username.
`git config --global user.email "your@email.com"`	Set your Git email.
`git init`	Initialize a Git repository in a folder.
`git clone <repository_url>`	Clone a repository from GitHub.
`git status`	Check the status of files (modified, new, staged).
`git add <filename>`	Add a specific file to staging.
`git add .`	Add all files to staging.
`git commit -m "Commit message"`	Save changes to Git.
`git push origin main`	Upload changes to GitHub.
`git pull origin main`	Download the latest changes from GitHub.
`git branch <branch-name>`	Create a new branch.
`git checkout <branch-name>`	Switch to another branch.
`git merge <branch-name>`	Merge a branch into the main branch.

GitHub and Repositories

What is a Repository?

A repository (repo) is a **storage location** where a project's files and history are kept.

Types of Repositories

1. **Local Repository** – Stored on your computer.
2. **Remote Repository** – Hosted on GitHub or another platform.

Public vs Private Repositories

- **Public Repository** – Anyone can view and contribute.
- **Private Repository** – Only authorized users can access it.

Creating a GitHub Repository

1. Go to GitHub and sign in.
2. Click on **New Repository**.
3. Enter a **repository name**.
4. Choose **Public** or **Private**.
5. Click **Create Repository**.

Connecting a Local Repository to GitHub

After creating a local Git repository, you can **link it to GitHub**:

```
git remote add origin <repository_url>
git branch -M main
git push -u origin main
```

Git Commits and Pushing Changes

What is a Commit?

A **commit** is a saved change in the project's history. It allows tracking modifications over time.

Steps to Commit and Push Changes

1. Check changes:

```
git status
```

2. Add files to staging:

```
git add .
```

3. Commit the changes with a message:

```
git commit -m "Updated project files"
```

4. Push the changes to GitHub:

```
git push origin main
```

Branching in Git

What is a Branch?

A **branch** allows developers to work on new features without affecting the main codebase.

Common Git Branch Commands

Command	Description
`git branch`	View all branches.
`git branch <branch-name>`	Create a new branch.
`git checkout <branch-name>`	Switch to another branch.
`git merge <branch-name>`	Merge a branch into the main branch.
`git branch -d <branch-name>`	Delete a branch.

Pull Requests and Merging

A **pull request (PR)** is a request to merge changes from one branch to another in GitHub.

Steps to Create a Pull Request in GitHub

1. Push changes to GitHub.
2. Open the repository on GitHub.
3. Click **Pull Requests > New Pull Request**.
4. Select the branches to merge.
5. Add a title and description.
6. Click **Create Pull Request**.

Git Ignore (.gitignore)

The `.gitignore` file tells Git which files **not to track**.

Common Files to Ignore

```
node_modules/
.env
__pycache__/
*.log
```

Add this file to prevent unnecessary files from being committed.

Undoing Changes in Git

Command	Description
`git checkout -- <file>`	Undo changes to a file.
`git reset HEAD <file>`	Unstage a file.
`git revert <commit-hash>`	Revert a commit.
`git reset --hard <commit-hash>`	Reset to a previous commit (dangerous).

Cloning a Repository

To copy an existing GitHub repository:

```
git clone <repository_url>
```

Conclusion

Git and GitHub are essential tools for version control in web development. They help in:

- ☑ **Tracking changes** in code.
- ☑ **Collaborating** with others.
- ☑ **Managing branches** for new features.
- ☑ **Hosting code** online using GitHub.