**Git:**

Git is a distributed version control system for tracking changes in source code during software development. It is designed for coordinating work among programmers, but it can be used to track changes in any set of files.

Version control, also known as source control, is the technique of tracking and managing changes to codes and these are the systems that are software tools that enable software teams to manage modifications to source code as time passes.

**GitHub:** GitHub is a "hub" (a place or platform) where Git users build software together. GitHub is also an hosting provider and version control platform you can use to collaborate on open source projects and share files.

**Differences between Git and GitHub**

Git is a version control system that manages and keeps track of your code. GitHub, on the other hand, is a service that let you host, share, and manage your code files on the internet.

**Benefits of Using Git**

 **History Tracking**: Git allows you to track every change made in your project, including: who made the change and when it was made.
 **Collaboration**: Multiple developers can be able work on the same project at the same time, and Git efficiently manages the merging of changes in code.
 **Branching and Merging**: Git enables developers to create branches to work on new features or bug fixes and later merge them back into the main codebase.
 **Offline Work**: Git works offline, which means you can commit changes and work on your project even without an internet connection.

**How to use Git and GitHub**

1. Install Git - **git –version -** Shows the current version of your Git.
2. Create a GitHub account.
3. Connect GitHub account to your Git account.

To set your Git username, type this in your terminal:

```
git config --global user.name "Swetha"
```

To set your Git email, type this in your terminal:

```
git config --global user.email "youremail@gmail.com"
```

You will be asked to authenticate your GitHub account, so just sign in with the same email to confirm.

4. Create your local files.
5. Create new repository in the github
6. **Push your local file to github.**
   **git init** - lets you initialize Git in your folder.
   **git remote add origin** https://github.com/swetha556/Practice.git **-** finally connects the local folder to the repository on GitHub. It is followed by the repository's link.
   **git add .** - lets you add all files in the present folder.
   **git commit -m "first commit"** - stores the added files. Use -m for message followed by the actual message.
   **git push -u origin main** - pushes the code to GitHub. The -u flag creates a tracking reference for the branch, and origin main puts the code in the main branch.
7. **Git repository to local machine.**
   **git clone <repo-URL> -** copy existing git repository in the local machine.
   **cd Practice**