**Git and GitHub**

# Introduction

* Git is a modern and widely used **distributed version control** system in the world. It is developed to manage projects with high speed and efficiency. The version control system allows us to monitor and work together with our team members at the same workspace.

# What is Git?

* **Git** is an **open-source distributed version control system**. It is designed to handle minor to major projects with high speed and efficiency. It is developed to co-ordinate the work among the developers. The version control allows us to track and work together with our team members at the same workspace.

# Features of Git

* **Open Source** :Git is an **open-source tool**.
* **Scalable**:Git is **scalable**, which means when the number of users increases, the Git can easily handle such situations.
* **Distributed**: Distributed means that instead of switching the project to another machine, we can create a "clone" of the entire repository.
* **Security**:Git is secure. It uses the **SHA1 (Secure Hash Function)** to name and identify objects within its repository. Files and commits are checked and retrieved by its checksum at the time of checkout.
* **Speed**:Git is very **fast**, so it can complete all the tasks in a while. Most of the git operations are done on the local repository, so it provides a **huge speed**. Also, a centralized version control system continually communicates with a server somewhere.
* The **core part of Git**is **written in C.**
* **Supports non-linear development**:Git supports **seamless branching and merging**, which helps in visualizing and navigating a non-linear development.
* **Branching and Merging**:**Branching and merging** are the **great feature**s of Git, which makes it different from the other SCM tools. Git allows the **creation of multiple branches** without affecting each other.
* **Staging Area**:The **Staging area** is also a **unique functionality** of Git. It can be considered as a **preview of our next commit**, moreover, an **intermediate area** where commits can be formatted and reviewed before completion.
* **Staging Area**:The **Staging area** is also a **unique functionality** of Git. It can be considered as a **preview of our next commit**, moreover, an **intermediate area** where commits can be formatted and reviewed before completion.



# Benefits of Git

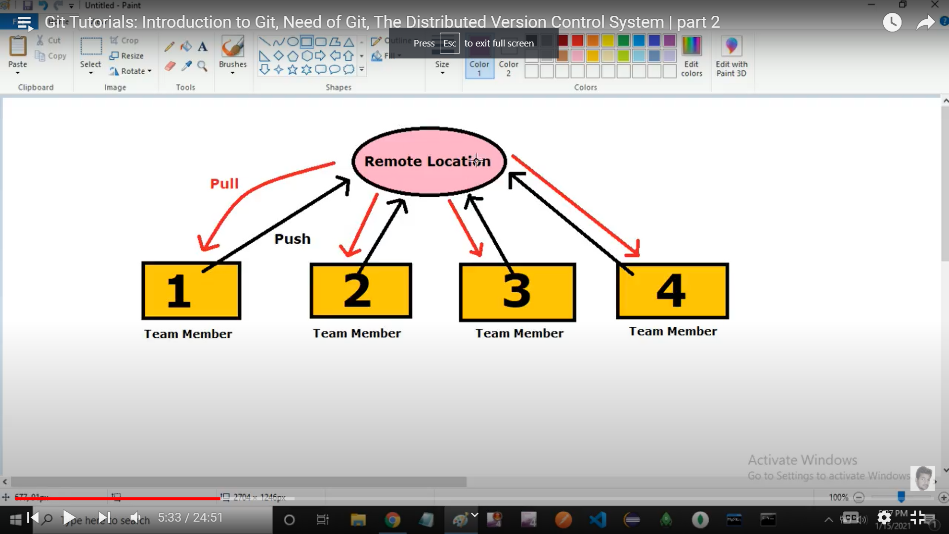
* A version control application allows us to **keep track** of all the changes that we make in the files of our project. Every time we make changes in files of an existing project, we can push those changes to a repository. Other developers are allowed to pull your changes from the repository and continue to work with the updates that you added to the project files.



* **Saves Time**:Git is lightning fast technology. Each command takes only a few seconds to execute so we can save a lot of time
* **Offline Working**:One of the most important benefits of Git is that it supports **offline working**. If we are facing internet connectivity issues, it will not affect our work.
* **Undo Mistakes**One additional benefit of Git is we can **Undo** mistakes.
* **Track the Changes**:Git facilitates with some exciting features such as **Diff, Log,** and **Status**, which allows us to track changes so we can **check the status, compare** our files or branches.

# Features of GitHub

* GitHub is a place where programmers and designers work together. They collaborate, contribute, and fix bugs together.
* **Some of its significant features are as follows.**
* Collaboration
* Integrated issue and bug tracking
* Graphical representation of branches
* Git repositories hosting
* Project management
* Team management
* Code hosting
* Track and assign tasks
* Conversations
* **The key benefits of GitHub are as follows.**
* It is easy to contribute to open source projects via GitHub.
* It helps to create an excellent document.
* You can attract recruiter by showing off your work. If you have a profile on GitHub, you will have a higher chance of being recruited.
* It allows your work to get out there in front of the public.
* You can track changes in your code across versions.



List of commands

1) git config –list { isse hume list mil jayega}

2) git config –global user.name ”github ka username”

3) git config –global user.email ”github ka emai lid”

4) apna username detail janne ke liye **git config user.name** aisa likhne se hume username mil jayega

5) apna email detail janne ke liye **git config user.email** aisa likhne se hume email id mil jayega

6) file create karne ke liye **touch file1.txt** karne se ek file1 name ka ek file create ho jayega

7) **git status** command se hume apne file ka status malum pdta hai

8) ek empty reposetory create karne ke liye hum **git init** command likhte hai yeh kafi important hai

9) untracked file to track karne ke liye hum **git add .** command likhte hai

10) apne file ko remote reposetory (github) me rakhne ke turant pahle usse commit karna hota hai **git commit –m “this is my first commit**”  
11)commit command hone ke baad hi hum usse github me store kar skte hai and usske liye hum **git remote add origin ”repo ka link”**

12) ab finally uss file ko apne github ke repo me dalna hota hai to hum kisi branch ko dalte hai or uska command hai **git push – u origin master**

13)

**1) Create a Repository for a Blank (New) Project**

->git init

**2) To create a file, run the cat or touch command as follows**

-> touch <file Name>

the list of all untracked files is displayed by the git status command.