**Git And GitHub**

**Git -**

* Git is version control system and tool that helps multiple people work on the same project without overwriting each others work
* It is a tool that helps us to keep track of changes in our files.
* It's very helpful when we are working with others on the same project
* We can see what changes were made to our files, who made them and when
* We don’t need an internet connection to use git
* Multiple people can work on the same project
* Git makes sure our files are safe

**GitHub -**

* GitHub is a website where we can store our code online.
* It helps us to and others work together on the same project
* We can save our code so we can access it from anywhere
* We can easily share our code and work with others on the same project
* We can keep track of bugs and tasks that need to be done
* We can write instructions and information about our project
* We create a account on GitHub and create a new repository on GitHub and make our first commit
* Fill in repository details
* Enter a name for our repository, Description,choose public / private,check box to add a README file
* Click on the create repository button
* Repository is created we add the changes and commit changes
* **README File -**
* A README file is a document that explains important details about a project. Its usually named “README.md'' or “README”
* README file describes what the project is about and its purpose
* Describes instructions on how to use the project
* **Commit -**
* A commit is a way to save changes we have made to files in our project
* It takes a snapshot of the changes we have made so we can keep track of them
* When we commit we are saving our latest changes to the project
* Each commit includes a message that explain what we changes
* Commits create history of our work so we can look back and see what we have done
* **Cloning in Github -**
* Cloning in GitHub is the process of creating a copy of a repository from GitHub to our local machine
* This allows us to work on the project on our own computer
* we can work on the work project files on our computer
* we can track changes
* Multiple people can work on the same project and merge their changes

1. First we open the repository page we want to clone for ex - “<https://github.com/divyayerkewar/JUnitassignment>”
2. Click the green “code” button and copy the URL
3. Then we open Git Bash
4. Use “cd” to navigate to the folder where we want to save the project
5. Use the “git clone” copy and and paste the URl we copied for ex - “ git clone “[https://github.com/divyayerkewar/JUnitassignment.git](https://github.com/divyayerkewar/JUnitassignment)”

* Cloning a repository copies it from github to our computer so we can work on it locally
* **Git Status -**
* git status command is used to check the current state of our repository
* It shows us which changes have been made, which files are staged for a commit and which files are not being tracked
* git status track files
* To use first we open the git bash and navigate to our project
* and Run “git status”
* **output** - On branch main Your branch is up-to-date with 'origin/main'. Untracked files: (use "git add <file>..." to include in what will be committed) abc.html nothing added to commit but untracked files present (use "git add" to track)
* In this example “abc.html” is an untracked file

**Types:**

1. **Untracked Files :** These are files that are in our working directory but are not being tracked by Git. They have not been added to the repository yet

**ex -** we created a new file name “myfile.txt” but have not added it to git.

**input** - git status

**output** - Untracked files: (use "git add

<file>..." to include in what will be committed) myfile.txt

1. **Modified Files** : These are files that have been changed in our working directory but have not been staged for commit

**ex** - we edited an existing file named “myfile.txt”

**input** - git status

**output** - Changes not staged for commit: (use "git add <file>..." to update what will be committed) (use "git checkout -- <file>..." to discard changes in working directory) modified: myfile.txt

**3.Unmodified Files** : These are files that have not been changed since the last commit.

**4.Staged Files** : These are files that have been added to the staging area and are ready to be committed.

**ex** - we added changes in “abc.txt” to the staging area

**input** - git add abc.txt

git status

**output** - Changes to be committed:

(use "git reset HEAD <file>..." to unstage)

modified: abc.txt

* **“git add” command -**
* git add command is used to add changes in our working directory to the staging area
* this prepares the changes to be included in the next commit
* git add command add new files to the repository
* and stage modification to existing files
* suppose we create a file named “hello.txt”
* we check the git status to see untracked files.

* **output** - Untracked files:

(use "git add <file>..." to include in what will be committed)

hello.txt

* use “git add” to stage the new file

git add hello.txt

* check status again

git status

* **output** - Changes to be committed:

(use "git reset HEAD <file>..." to unstage)

new file: hello.txt

* **“git push” command -**
* git push command is used to upload local repository content to a remote repository
* pushing means we transfer commits from our local repository to aremote repository on github
* first we ensure that local repository is linked to the remote reposiotry on github
* push the commit changes to the remote repository
* **input** - git push origin main
* **origin** - origin is the default name for the remote repository
* **main** - main is the branch we are pushing to
* git status
* **output** - On branch main

Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean