SAI SUBHAMAYA DAS

2141004165

Create a tutorial for Git and GitHub with commands.

What is GitHub?

- GitHub is a **website** where you can **store your Git projects online**. It’s like **Google Drive for code**, but with extra tools that make it easy to:

* Share your work with others
* Collaborate with teammates
* Show your projects to the world

What is Git ?

Git is a **tool** that helps you **track changes** in your code or files over time. Think of it like the **undo/redo** feature in Word or Google Docs, but way more powerful. It helps you:

* Save different versions of your work
* Go back to older versions if something breaks
* Work with others on the same project without messing each other up

Installation

Download Git from:

<https://git-scm.com/downloads>

To verify installation: git --version

SETUP

Configuring user information used across all local repositories

1. **git config --global user.name “[firstname lastname]”**

- set a name that is identifiable for credit when review version history.

**2.** **git config --global user.email “[valid-email]”**

-set an email address that will be associated with each

history marker.

SETUP & INIT

Configuring user information, initializing and cloning repositories.

**1. git init**

**-**initialize an existing directory as a Git repository.

**2. git clone [url]**

-retrieve an entire repository from a hosted location via

url

**STAGE & SNAPSHOT**

-Working with snapshots and the Git staging area

**1.** **git status**

-show modified files in working directory, staged for your next commit

**2.** **git add [file]**

-add a file as it looks now to your next commit (stage)

**3.** **git commit -m “[descriptive message]”**

-commit your staged content as a new commit snapshot

BRANCH & MERGE

-Isolating work in branches, changing context, and integrating changes

**1.** **git branch**

-list your branches. a \* will appear next to the currently active branch

**2.** **git branch [branch-name]**

-create a new branch at the current commit

**3.** **git checkout**

-switch to another branch and check it out into your working directory

**4. git merge[branch]**

-merge the specified branch’s history into the current one

**5. git log**

- show all commits in the current branch’s history

TRACKING PATH CHANGES

**1. git rm [file]**

-delete the file from project and stage the removal for commit

**2.** **git branch -d branch\_name**

-Delete Branch

SHARE & UPDATE

Retrieving updates from another repository and updating local repos

**1.git pull**

-fetch and merge any commits from the tracking remote branch