**Assignments 2 Swapnil Gaikwad**

1. What is the purpose of git clone?

Answer - The purpose of the git clone command is to create a local copy of an existing remote Git repository. It copies the entire repository, including all of its files, history, branches, and commits, onto your local machine, making it ready for you to work on.

1. Can you explain the difference between git clone and git pull?

Answer -

git clone: You use it once, when setting up a project locally for the first time. It copies the entire history and files of a repository from the remote to your local system and sets up a connection to the remote repository for future interactions.

git pull: It retrieves the latest changes from the remote (new commits, branches, etc.) and merges them into your current working branch. You use it frequently, whenever you want to sync your local repository with the remote.

1. How would you authenticate your Git account using a Personal Access Token (PAT)?

Answer -

Generate the PAT from GitHub. GitHub Personal Access Tokens.

Click Generate token then select the scopes (permissions) you need, like repo, workflow, etc., depending on your needs and copy the token immediately, as you won’t be able to view it again.

In your terminal, when pushing changes, you’ll be prompted like this:

git push origin main

Username for 'https://github.com': username

Password for 'https://username@github.com': <Paste your PAT here>

1. How do you add multiple files to the staging area at once?

Answer -

Using below commands you can add multiple files to staging area.

Git add .

Git add \*

Git add file1.txt file2.txt file3.txt

1. What does the command git add . do?

Answer -

git add . stages all modified and untracked files in the current directory and its subdirectories for the next commit in a Git repository.

1. Explain the purpose of the .gitignore file and provide examples of its usage.

Answer –

The .gitignore file is used to specify which files or directories Git should ignore when you commit changes to a repository. It helps to avoid tracking files that are not necessary or should not be versioned, such as temporary files, log files, build artifacts, sensitive information (like passwords), or IDE-specific files.

Purpose of the .gitignore file:

Prevent unnecessary files from being tracked:

Protect sensitive information:

Reduce repository size:

Prevent conflicts:

Example of a .gitignore file:

# Ignore all .log files

\*.log

# Ignore build directory

/build/

# Ignore IDE configurations

.vscode/

/.idea/

# Don't ignore a specific log file

!important.log

# Ignore OS-specific files (e.g., macOS Finder, Windows Thumbnails)

.DS\_Store

Thumbs.db