Git Tutorial

MD. Abdullah All Mahmud and Jubayer Ahamed

Git is a free and open-source distributed version control system that tracks changes to files and allows for collaboration on projects. It's a fundamental tool for software development, enabling users to manage different versions of their code, track who made which changes and revert to previous states if needed. Git works on your computer, but you also use it with online services like GitHub, GitLab, or Bitbucket to share your work with others.
Suppose a university software team is developing a **Library Management System** to manage books, track student borrowing, and calculate fines. The project has modules like Book Registration, Student Information, and Fine Calculation. Since new features are added frequently, proper version control is needed to keep track of changes and return to a stable version if required.
The team is using Git locally. They initialize the project with Git Bash, add and commit files with clear messages after every change. For example, if the Book Registration module is updated, they commit it with a note explaining the change. This helps maintain a clear project history and supports teamwork during development.

Core Concepts:

- Repository: A directory where Git tracks all project files and their history.
- Clone: Creating a local copy of a remote repository.
- Commit: Saving a snapshot of changes to the repository with a descriptive message.
- Branch: Creating separate lines of development to work on features or bug fixes independently.
- Merge: Combining changes from different branches into one.
- Push: Uploading local commits to a remote repository.
- Pull: Fetching changes from a remote repository and integrating them into the local repository.
- Staging Area: A temporary area where you stage changes before committing them.
- Remote Repository: A centralized repository hosted online (e.g., on GitHub, GitLab, Bitbucket).

Basic Git Commands:

- git init: Initializes a new Git repository in the current directory.
- git push: Uploads local commits to a remote repository.
- git pull: Fetches changes from a remote repository and merges them.
- git commit -m "Your commit message": Saves staged changes to the repository.
- git branch: Lists, creates, or deletes branches.
- git clone <repository_url>: Clones a remote repository to your local machine.
- git status: Shows the current state of the working directory and staging area.
- git add <file>: Stages a file for the next commit.
- git add . or git add -A: Stages all changes in the current directory.
- git checkout <branch_name>: Switches to a different branch.
- git merge <branch_name>: Merges changes from a specified branch into the current branch.
- git log: Displays the commit history.
- git diff: Shows the differences between files.

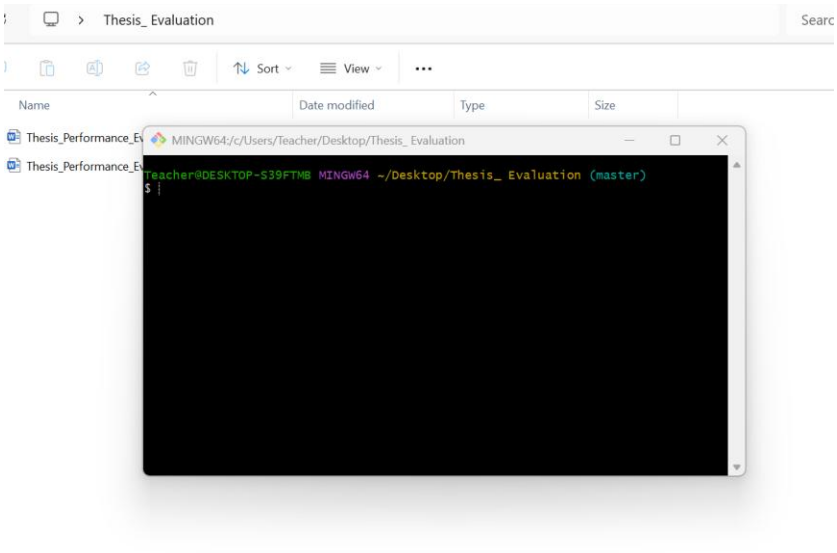
Workflow Example:

1. Initialize: Start a new project or clone an existing one.
2. Add Files: Use git add to stage files you've modified or created.
3. Commit Changes: Use git commit to save the staged changes with a descriptive message.
4. Push Changes: Use git push to upload your local commits to a remote repository.
5. Collaborate: Use branches, pull requests, and merges to work with others.

Adding a file to a repository

Follow the steps below to practice the basic Git commands on your computer:

- **Go to the folder then Open the Git Bash**



- **Configure User Info (only first time per computer)**

git config --global user.name "Your Name"

git config --global user.email "your_email@example.com"

- **Initialize Git**

git init

```
Teacher@DESKTOP-S39FTMB MINGW64 ~/Desktop/Thesis_ Evaluation
$ git init
Initialized empty Git repository in C:/Users/Teacher/Desktop/Thesis_ Evaluation/
.git/
```

- **Add origin**

git remote add origin GithubRepositiryLink

```
Teacher@DESKTOP-S39FTMB MINGW64 ~/Desktop/Thesis_ Evaluation (master)
$ git remote add origin https://github.com/Jubayer530/Files.git
```

- **Add the File**

```
Teacher@DESKTOP-S39FTMB MINGW64 ~/Desktop/Thesis_ Evaluation (master)
$ git add .
```

- **Commit the file that you've staged in your local repository**

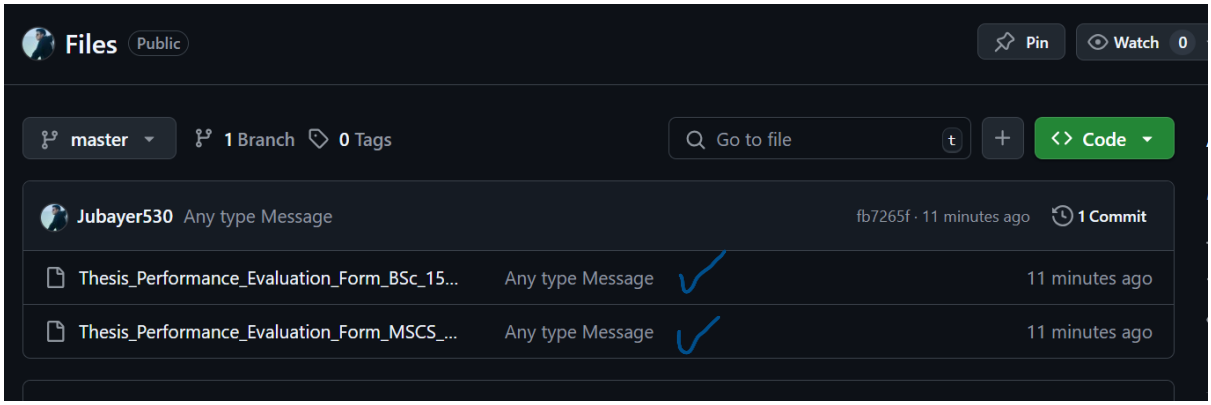
git commit -m "Any type of Message"

```
Teacher@DESKTOP-S39FTMB MINGW64 ~/Desktop/Thesis_ Evaluation (master)
$ git commit -m "Any type Message"
[master (root-commit) fb7265f] Any type Message
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 Thesis_Performance_Evaluation_Form_BSc_1569228630.docx
 create mode 100644 Thesis_Performance_Evaluation_Form_MSCS_1569229796 (1).docx
```

- **Push the files to the repository**

```
Teacher@DESKTOP-S39FTMB MINGW64 ~/Desktop/Thesis_ Evaluation (master)
$ git push -u origin master
```

After a successful push, then go to your repository and click on refresh:



Lab Task:

Create a GitHub repository named GitLabTask. It must contain:

- README file
- Upload the file (from your desktop's folder) to the repository using git commands
- At least one commit / One updated message

Then, share the link of your GitHub repository.

References:

1. https://www.geeksforgeeks.org/git/git-tutorial/
2. https://www.w3schools.com/git/
3. https://git-scm.com/book/en/v2/Git-Basics-Getting-a-Git-Repository