GIT COMMAND

1.GIT INIT: The git init command is used to create a new blank repository. It is used to make an existing project as a Git project. Several Git commands run inside the repository, but init command can be run outside of the repository.

```
SER@EMON MINGW64 /e/Third Year odd Semester/Software Engineering & Information
ystem Design/LAB/lab-3 (master)
 git init
einitialized existing Git repository in E:/Third Year odd Semester/Software Eng
neering & Information System Design/LAB/lab-3/.git/
```

2.GIT CONFIG: The GIT CONFIG command is a convenience function that is used to set git configuration values on global or local project level

```
USER@EMON MINGW64 /e/Third Year odd Semester/Software Engineering & Information
System Design/LAB/lab-3 (master)
$ git config --global user.name "Morsed Emon"

USER@EMON MINGW64 /e/Third Year odd Semester/Software Engineering & Information
System Design/LAB/lab-3 (master)
$ git config --global user.mail "morseduremonece@gmail.com"
```

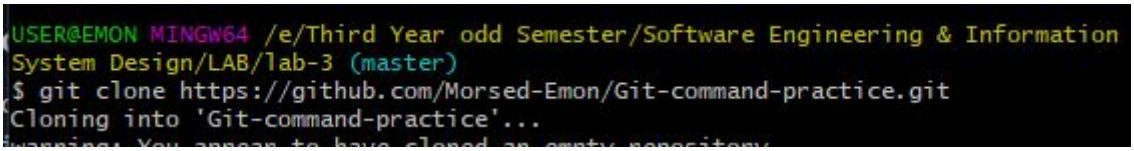
3.GIT ADD: The git add command is used to add file contents to the Index. This command updates the current content of the working tree to the staging area. It also prepares the staged content for the next commit. Every time we add or update any file in our project, it is required to forward updates to the staging area.

```
USER@EMON MINGW64 /e/Third Year odd Semester/Software Engineering & Information
System Design/LAB/lab-3 (master)
$ git add --all
```

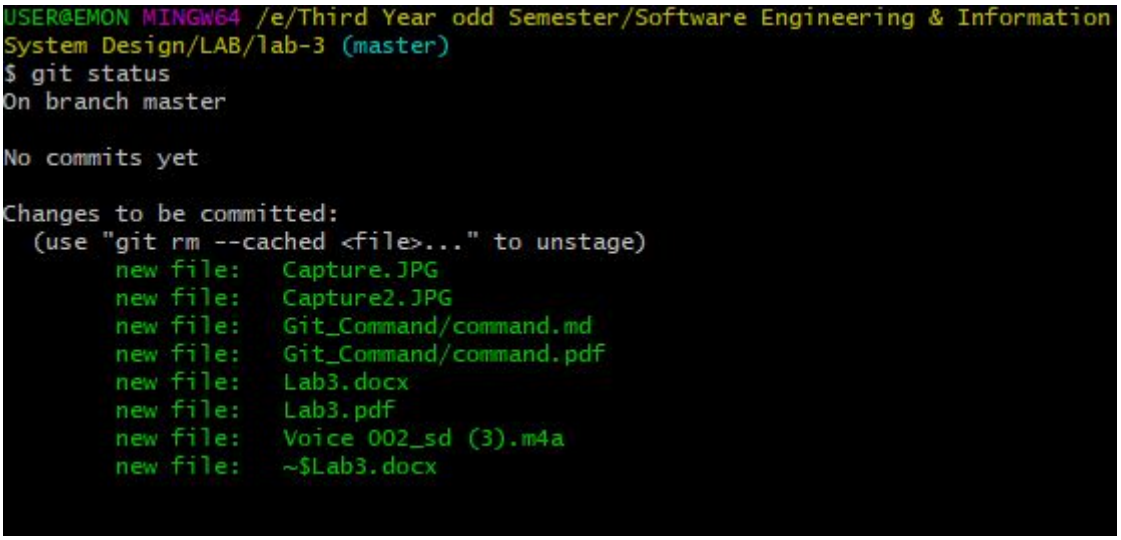
4.GIT COMMIT: It is used to record the changes in the repository. It is the next command after the GIT ADD. Every commit contains the index data and the commit message. Every commit forms a parent-child relationship. When we add a file in Git, it will take place in the staging area. A commit command is used to fetch updates from the staging area to the repository. Commits are the snapshots of the project.

```
USER@EMON MINGW64 /e/Third Year odd Semester/Software Engineering & Information
System Design/LAB/lab-3 (master)
$ git commit -m "SOFTWARE LABTASK"
[master (root-commit) b3a5654] SOFTWARE LABTASK
 8 files changed, 10 insertions(+)
 create mode 100644 Capture.JPG
 create mode 100644 Capture2.JPG
 create mode 100644 Git_Command/command.md
 create mode 100644 Git_Command/command.pdf
 create mode 100644 Lab3.docx
 create mode 100644 Lab3.pdf
 create mode 100644 Voice 002_sd (3).m4a
 create mode 100644 ~$Lab3.docx
```

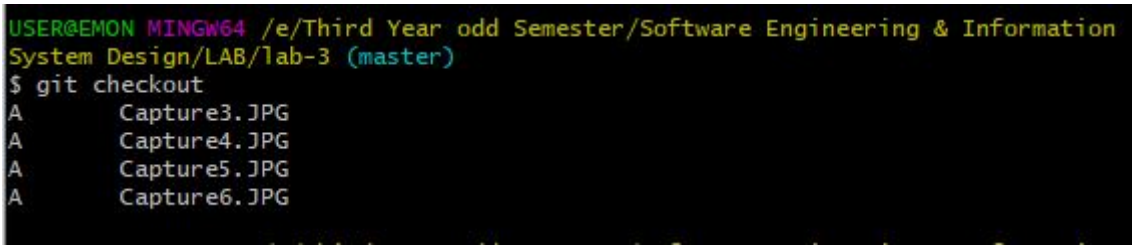
5.GIT CLONE: The git clone is a command-line utility which is used to make a local copy of a remote repository. It accesses the repository through a remote URL. Usually, the original repository is located on a remote server, often from a Git service like GitHub, Bitbucket, or GitLab. The remote repository URL is referred to the origin.



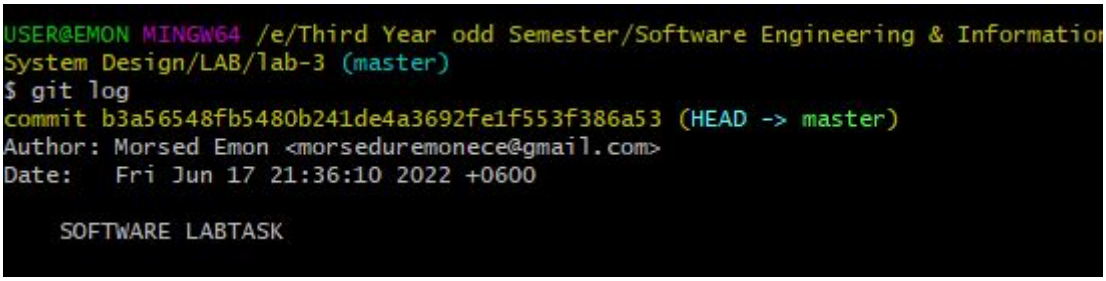
6.GIT STATUS: The git status command displays the state of the working directory and the staging area. It lets you see which changes have been staged, which haven't, and which files aren't being tracked by Git. Status output does not show you any information regarding the committed project history.



7.GIT CHECKOUT: In Git, the term checkout is used for the act of switching between different versions of a target entity. The git checkout command is used to switch between branches in a repository.



8.GIT LOG: Git log is a utility tool to review and read a history of everything that happens to a repository. Multiple options can be used with a git log to make history more specific.



9.GIT PULL: The term pull is used to receive data from GitHub. It fetches and merges changes from the remote server to your working directory. The git pull command is used to pull a repository.

10.GIT PUSH: The push term refers to upload local repository content to a remote repository. Pushing is an act of transfer commits from your local repository to a remote repository. Pushing is capable of overwriting changes; caution should be taken when pushing.