**A1**. Demonstrate and Create project in local and remote repository using GitBash and GitHub and apply  init, status, log, add, commit, push, config, clone and reset commands on repository.

Git **init** command:

Description

git init is used to initialize a new Git repository. It sets up the necessary structure for tracking the history of the project.

Syntax : git init

Example : $ git init

Initialized empty Git repository in /path/to/repository/.git/

Git **status** command:

Description

git status shows the current state of the repository. It displays which files are staged, unstaged, or untracked.

Syntax : git status

Example : $ git status

On branch main

nothing to commit, working tree clean

Git **log** command:

Description

git log shows the commit history for the repository. It displays detailed information about each commit, such as the commit hash, author, date, and message.

Syntax : git log

Example : $ git log

commit a1b2c3d4e5f67890

Author: John Doe <john@example.com>

Date: Mon Sep 20 14:22:34 2025 -0700

Initial commit

Git **add** command:

Description

git add adds changes in the working directory to the staging area. You must use this command before committing changes to the repository.

Syntax : git add <file> git add .

Example : $ git add README.md

Git **commit** command:

Description

git commit takes the changes staged with git add and commits them to the repository with a message describing the changes.

Syntax : git commit -m "Your commit message"

Example : $ git commit -m "Added README file"

[main (root-commit) a1b2c3d] Added README file

1 file changed, 12 insertions(+)

create mode 100644 README.md

Git **push** command:

Description

git push sends your local commits to the remote repository (e.g., GitHub). This updates the remote repository with your changes.

Syntax : git push <remote> <branch>

Example : $ git push origin main

Git **config** command:

Description

git config is used to set configuration options for Git, such as user name, email, or preferred editor.

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

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

Example : $ git config --global user.name "John Doe"

$ git config --global user.email "john@example.com"

Git **clone** command:

Description

git clone is used to copy an existing remote repository to your local machine. This command creates a new directory with the repository’s files and history.

Syntax : git clone <repository\_url>

Example : $ git clone https://github.com/username/repository.git

Cloning into 'repository'...

remote: Enumerating objects: 100, done.

remote: Counting objects: 100% (100/100), done.

remote: Compressing objects: 100% (40/40), done.

Receiving objects: 100% (100/100), done.

Resolving deltas: 100% (60/60), done.

Git **Reset** command:

Description

git reset is used to undo changes. It can unstage files or reset the commit history to a previous state.

Syntax : git reset <commit>

Example : git reset HEAD~1

git reset --hard

**Git user configuration details**

git config --global user.name "Gauthams123"

git config --global user.email [vvce23cse0062@vvce.ac.in](mailto:vvce23cse0062@vvce.ac.in)

**Create a local repository** using git init:

mkdir my-project

cd my-project

git init

**Create a GitHub repository** via GitHub's website (create a new repository).

**Link your local repository to the remote repository** using git remote add origin:

git remote add origin <https://github.com/username/my-project.git>

**Add files to the local repository**, for example:

touch README.md

git add README.md

**Commit your changes**:

git commit -m "Initial commit"

**Clone the repository** to another local machine or directory (if needed):

git clone <https://github.com/username/my-project.git>

**Use git status and git log** to track changes and view history.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Configuration** | **Execution** | **Viva** | **Total** | **Verified By** |
|  |  |  |  |  |