Experiment no.2

title: Implementation of Stagging, Commit and Push on GitHub

Theory :

To perform Version Control using GIT. Git is a free and open-source distributed version control system designed to handle everything from small to very large projects with speed and efficiency. Git is easy to learn and has a tiny footprint with lightning-fast performance. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like cheap local branching, convenient staging areas, and multiple workflows. Some of the basic operations in

Git are: 1. Initialize

2.Add

3.Commit

4.Pull

5.Push

Some advanced Git operations are:

1.Branching

2.Merging

3.Rebasing

The following diagram depict the all supported operations in GIT

Installation of GIT

1.Inwindows,downloadGITfromhttps://git-scm.com/andperformthe straightforward installation.

2.In Ubuntu, install GIT using $sudo apt install git, Confirm the version after installation using command $git –version



Git and GitHub are two of the most popular tools used for version controland collaboration in software development.

Here are some common Git and GitHub commands:

Initializing a Git repository: $ git init

Checking the status of your repository: $ git status

Adding files to the stage: $ git add <file-name>

Committing changes: $ git commit -m "commit message"

Checking the commit history: $ git log

Undoing changes: $ git checkout <file-name>

Creating a new branch: $ git branch <branch-name>

Switching to a different branch: $ git checkout <branch-name>

Merging two branches: $ git merge <branch-name>

Pushing changes to a remote repository: $ git push origin <branch-name>

Cloning a repository from GitHub: $ git clone <repository-url>

Creating a pull request on GitHub: Go to the repository on GitHub,select the branch you want to merge and click the "New pull request"button.These are just a few of the many Git and GitHub commands available.

There are many other Git commands and functionalities that you can explore to suit your needs.

Description: To practice source code management on GitHub, you can follow these steps:

Create a GitHub account if you don't already have one.

Create a new repository on GitHub.

Clone the repository to your local machine: $ git clone <repository-url>

Move to the repository directory: $ cd <repository-name>

Create a new file in the repository and add the source code written in exercise 1.

Stage the changes: $ git add <file-name>

Commit the changes: $ git commit -m "Added source code for a simple user registration form"

Push the changes to the remote repository: $ git push origin master

Verify that the changes are reflected in the repository on GitHub. These steps demonstrate how to use GitHub for source code management. You can use the same steps to manage any source code projects on GitHub. Additionally, you can also explore GitHub features such as pull requests, code review, and branch management to enhance your source code management workflow.