Experiment 03 - Git Commands

Roll No.	70
Name	MAYURI SHRIDATTA YERANDE
Class	D15-B
Subject	DevOps Lab
LO Mapped	LO1: To understand the fundamentals of DevOps engineering and be fully proficient with DevOps terminologies, concepts, benefits, and deployment options to meet your business requirements LO2: To obtain complete knowledge of the "version control system" to effectively track changes augmented with Git and GitHub

Aim: To Perform various GIT operations on local and Remote repositories using GIT Cheat-Sheet

Introduction:

Git Commit: Takes the staged snapshot and commits it to the project history.

Git Pull: The term Pull is used to receive data from GitHub. It fetches and merges changes on the remote server to your working directory.

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.

Git Branch: A branch is a version of the repository that diverges from the main working project. It is an essential feature available in most modern version control systems. A Git project can have more than one branch.

Git Clone: The git clone is a Git command-line utility. It is used to make a copy of the target repository or clone it.

Git Fetch: It is used to fetch branches and tags from one or more other repositories, along with the objects necessary to complete their histories. It updates the remote-tracking branches.

Git Fork: A fork is a rough copy of a repository. Forking a repository allows you to freely test and debug with changes without affecting the original project.

Git checkout: In addition to checking out old commits and old file revisions, git checkout is also the means to navigate existing branches.

Git add: Moves changes from the working directory to the staging area. This gives you the opportunity to prepare a snapshot before committing it to the official history.
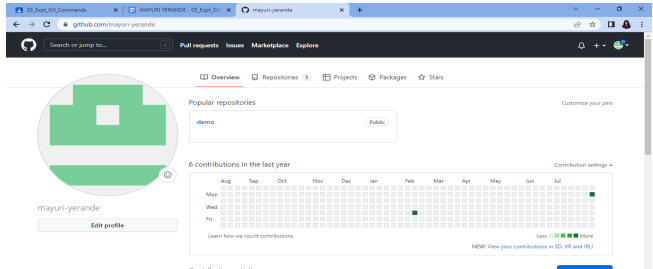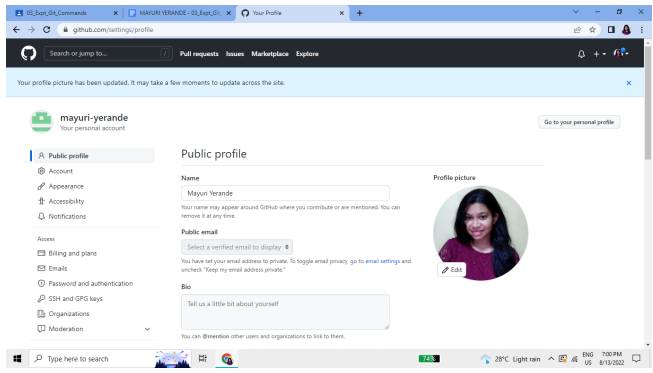
Git init: Initializes a new Git repository.

Git log: Lets you explore the previous revisions of a project. It provides several formatting options for displaying committed snapshots.

Git merge: A powerful way to integrate changes from divergent branches.

**Git status:** Displays the state of the working directory and the staged snapshot.

Just getting started – GitHub

Just using GitHub, please complete each task below. This requires that you have already set up an account.

Change your profile picture and name	*Before changing:* *Click on "Edit profile" and edit the profile pic and name from that section* *After changing:*
Create a repository named GitHubExpt03_grno	Go to "Repositories" and click on "new" Name your repository

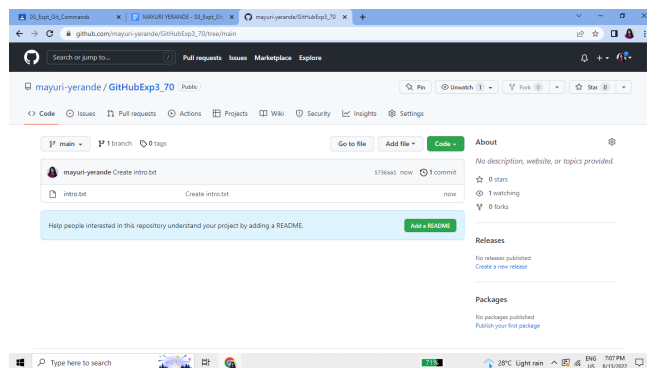## Repository has been created
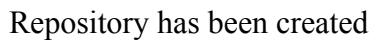
Create a file in the above directory named "intro.txt" with your name on the first line.
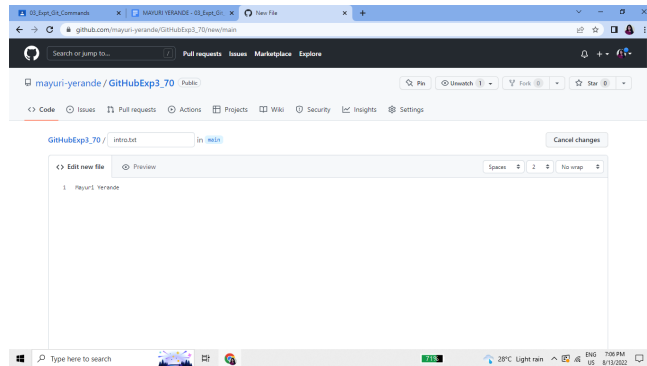
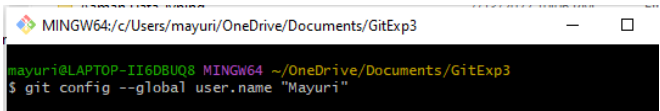*Commit the file*
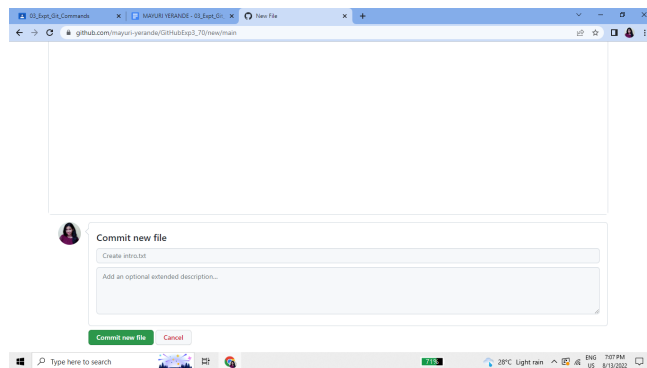
File has been created

To show ALL of above in one screen, you should be able to go to the URL (or close):

https://github.com/<your username>/GitHubExpt03_grno

Just getting started – Git Bash

Just using Git, please complete each task below and write the answer in the blank box beside the question. This requires that you have already installed Git.

setup user name (Please use your given name)	Open Git Bash. Type the command 'git config –global user.name "*username*" '. This will set up your username. You can check by the command 'git config –global user.name'. MINGW64:/c/Users/mayuri/OneDrive/Documents/GitExp3 — □ mayuri@LAPTOP-II6DBUQ8 MINGW64 ~/OneDrive/Documents/GitExp3 $ git config --global user.name "Mayuri"
setup user email (Please use your ves account)	Type the command 'git config –global user.email "*email-id*" '. This will set up your email. You can check by the command 'git config –global user.email mayuri@LAPTOP-II6DBUQ8 MINGW64 ~/OneDrive/Documents/GitExp3 $ git config --global user.email "2020.mayuri.yerande@ves.ac.in" mayuri@LAPTOP-II6DBUQ8 MINGW64 ~/OneDrive/Documents/GitExp3 $	
Create your local repository named GitExpt03_grno	Go to the directory where you want to create a local repository. Type the command 'git init "*repository_name*" '. This will create an empty repository. mayuri@LAPTOP-II6DBUQ8 MINGW64 ~/OneDrive/Documents/GitExp3 $ git init "GitExp03_70" Initialized empty Git repository in C:/Users/mayuri/OneDrive/Documents/GitExp3/GitExp03_70/.git/ mayuri@LAPTOP-II6DBUQ8 MINGW64 ~/OneDrive/Documents/GitExp3

Add a file in above local repository named file_grno_rollno and add some content to file
 	
Add it to staging state
	``
	mayuri@LAPTOP-II6DBUQ8 MINGW64 ~/OneDrive/Documents/GitExp3/GitExp03_70 (master)
	$ git status
	On branch master
	
	No commits yet
	
	Untracked files:
	  (use "git add <file>..." to include in what will be committed)
	        hello.txt
	
	nothing added to commit but untracked files present (use "git add" to track)
	``
	
	``
	mayuri@LAPTOP-II6DBUQ8 MINGW64 ~/OneDrive/Documents/GitExp3/GitExp03_70 (master)
	$ git add hello.txt
	``

Commit the files	To commit the files, type 'git commit -m "some message"
	
	``
	mayuri@LAPTOP-II6DBUQ8 MINGW64 ~/OneDrive/Documents/GitExp3/GitExp03_70 (master)
	$ git commit -m "First Commit"
	[master (root-commit) 58eb4a8] First Commit
	 1 file changed, 1 insertion(+)
	 create mode 100644 hello.txt
	``

Create a scenario for modifications/additions of the multiple files. Show the advantage of staging state.	Here, you can create multiple files. Add a few of them to the staging area with 'git add' and then look for the status using 'git status'. Then make a Commit. You can keep a track using 'git status' and 'git log'
	
	``
	mayuri@LAPTOP-II6DBUQ8 MINGW64 ~/OneDrive/Documents/GitExp3/GitExp03_70 (master)
	$ git status
	On branch master
	Untracked files:
	  (use "git add <file>..." to include in what will be committed)
	        second.txt
	
	nothing added to commit but untracked files present (use "git add" to track)
	
	mayuri@LAPTOP-II6DBUQ8 MINGW64 ~/OneDrive/Documents/GitExp3/GitExp03_70 (master)
	$ git add second.txt
	
	mayuri@LAPTOP-II6DBUQ8 MINGW64 ~/OneDrive/Documents/GitExp3/GitExp03_70 (master)
	$ git status
	On branch master
	Changes to be committed:
	  (use "git restore --staged <file>..." to unstage)
	        new file: second.txt
	
	mayuri@LAPTOP-II6DBUQ8 MINGW64 ~/OneDrive/Documents/GitExp3/GitExp03_70 (master)
	$ git commit -m "second"
	[master b80b517] second
	 1 file changed, 1 insertion(+)
	 create mode 100644 second.txt
	
	mayuri@LAPTOP-II6DBUQ8 MINGW64 ~/OneDrive/Documents/GitExp3/GitExp03_70 (master)
	$ git status
	``

Display the log	To see the log, just type the command 'git log'
	
	``
	mayuri@LAPTOP-II6DBUQ8 MINGW64 ~/OneDrive/Documents/GitExp3/GitExp03_70 (master)
	$ git log
	commit b80b517164347b74f0e2dd5ea5b4fcee5ebd8918 (HEAD -> master)
	Author: Mayuri <2020.mayuri.yerande@ves.ac.in>
	Date: Sat Aug 13 19:40:25 2022 +0530
	
	    second
	
	commit 58eb4a855bbb344777483f5bb283478119c01a9c
	Author: Mayuri <2020.mayuri.yerande@ves.ac.in>
	Date: Sat Aug 13 19:37:37 2022 +0530
	
	    First Commit
	``

Download your GitHub repository created in table-1 named GitHubExpt03_grno	To add your GitHub repository to the local machine, type command 'git remote add origin *link*'. The link you'll get on your GitHub repository section.
	
	`mayuri@LAPTOP-II6DBUQ8 MINGW64 ~/OneDrive/Documents/GitExp3/GitExp03_70 (master)`
	`$ git remote add origin https://github.com/mayuri-yerande/GitHubExp3_70.git`

Make some changes in the GitHub file.
 	
Pull the latest files from your GitHub repository
	
	`mayuri@LAPTOP-II6DBUQ8 MINGW64 ~/OneDrive/Documents/GitExp3/GitExp03_70 (master)`
	`$ git push -u origin master`
	
	`Enumerating objects: 6, done.`
	`Counting objects: 100% (6/6), done.`
	`Delta compression using up to 4 threads`
	`Compressing objects: 100% (3/3), done.`
	`Writing objects: 100% (6/6), 477 bytes | 22.00 KiB/s, done.`
	`Total 6 (delta 0), reused 0 (delta 0), pack-reused 0`
	`remote:`
	`remote: Create a pull request for 'master' on GitHub by visiting:`
	`remote:        https://github.com/mayuri-yerande/GitHubExp3_70/pull/new/master`
	`remote:`
	`To https://github.com/mayuri-yerande/GitHubExp3_70.git`
	` * [new branch]      master -> master`
	`branch 'master' set up to track 'origin/master'.`

Make some changes in the local file of GitHub repository
 	
Update the changes back to Github Repository
	Make some changes in the local file and commit the changes. Then type 'git push -u origin master'. Visit your GitHub, go into the commit history, you'll see the commit changes made.
	
	`mayuri@LAPTOP-II6DBUQ8 MINGW64 ~/OneDrive/Documents/GitExp3/GitExp03_70 (master)`
	`$ git pull origin`
	`remote: Enumerating objects: 12, done.`
	`remote: Counting objects: 100% (12/12), done.`
	`remote: Compressing objects: 100% (7/7), done.`
	`remote: Total 12 (delta 0), reused 0 (delta 0), pack-reused 0`
	`Unpacking objects: 100% (12/12), 2.60 KiB | 3.00 KiB/s, done.`
	`From https://github.com/mayuri-yerande/GitHubExp3_70`
	` * [new branch]      main        -> origin/main`
	`Already up to date.`

Team set up in GitHub

Conclusion :Commands like git add, git log, git status, git commit were used to perform operations on the files. Local and Remote repositories were connected and operations were performed using commands like git push, git pull. Collaboration for Projects was also performed using GitHub.