**Difference between git,github and gitbash:**

**Git**: A tool used to track changes in your code and manage versions.

**GitHub**: A website where you can store and share your Git repositories online.

**Git Bash**: A command-line tool for Windows to run Git commands.

**Commands lab1**

ls

Lists the contents of the current directory.

touch demo.java1

Creates an empty file named demo.java.

git init

Initializes a new Git repository in the current directory. Creates a . git folder to track version control.

ls -a

Lists all files in the directory, including hidden files .

git status

Shows the current state of the working directory and staging area. Tells you which files are tracked, untracked, or modified.

git add demo.java1

Adds demo.java to the staging area, preparing it for commit.

git commit -m "first commit"

Commits the staged changes with a message "first commit". This saves a snapshot of the project.

git branch -m main

Renames the current branch (usually master) to main. This is a common practice to align with modern Git naming conventions.

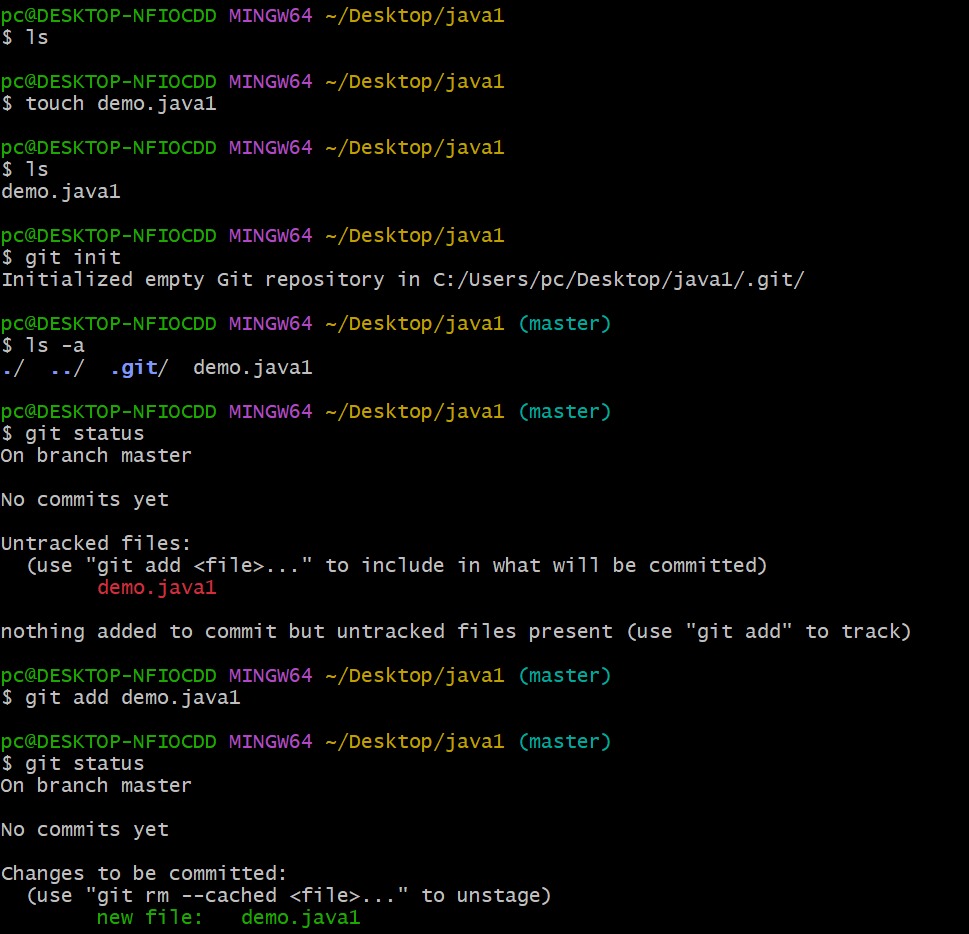
git remote add origin https://github.com/ananshi04/Ananshi.git

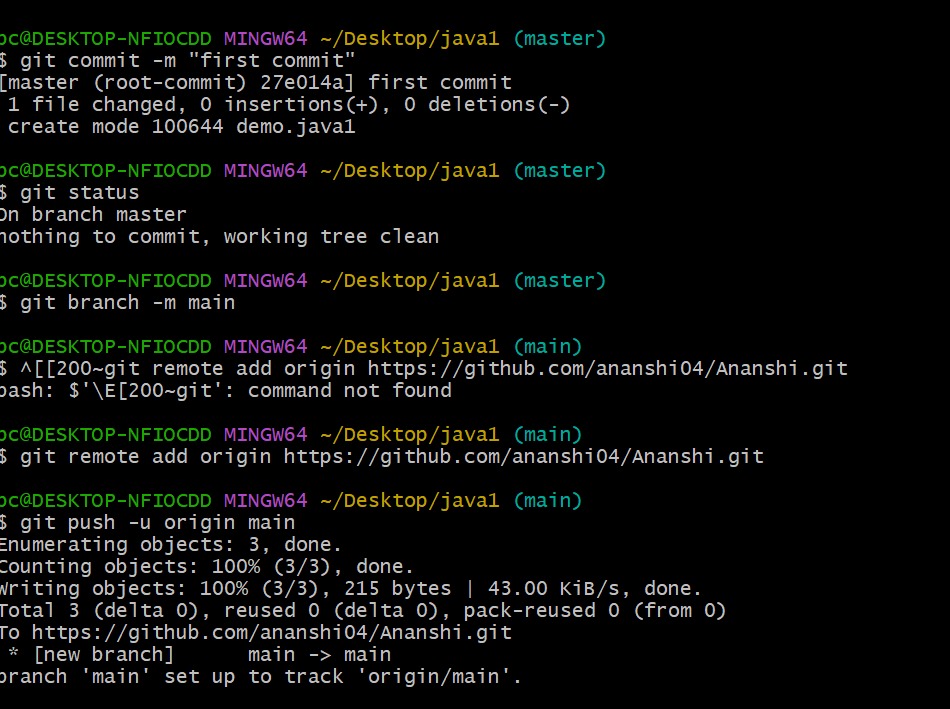
Adds a remote repository named origin pointing to your GitHub repo.

This lets you push and pull code between your local and remote repositories.

git push -u origin main

Pushes your local main branch to the remote origin and sets it as the default upstream branch for future pushes and pulls.





**Commands lab2**

**git branch**

**Use:  
Displays a list of all local branches in the Git repository.  
It shows which branch you are currently on with an asterisk (\*).**

**git switch <branch-name>**

**Use:  
Switches to the specified existing branch.  
For example, git switch abc moves you to the abc branch.**

**git checkout -b <branch-name>**

**Use:  
Creates a new branch and immediately switches to it.  
Example: git checkout -b xyz creates and switches to branch xyz.**

**git switch -c <branch-name>**

**Use:  
Another way to create and switch to a new branch (like checkout -b).  
Example: git switch -c branch3.**

**ls**

**Use:  
Lists files and folders in the current working directory (Unix/Linux style).  
Useful for checking which files are present.**

**git log**

**Use:  
Shows the commit history of the current branch.  
It displays commit ID, author, date, and message.**

**git checkout --orphan <branch-name>**

**Use:  
Creates a new branch with no commit history (a fresh branch).  
Useful for starting a completely new line of development.**

**touch <filename>**

**Use:  
Creates a new, empty file.  
Example: touch demo1.txt creates a file named demo1.txt.**

**git add <filename>**

**Use:  
Stages the specified file for the next commit.  
Example: git add demo1.txt prepares it for committing.**

**git commit -m "message"**

**Use:  
Creates a new commit with a short description.  
Example: git commit -m "add demo1.txt" saves the file changes with a message.**

**git checkout <branch-name>**

**Use:  
Switches to the specified branch (if it exists).  
Example: git checkout main moves to the main branch.**

**git push -u origin <branch-name>**

**Use:  
Pushes the branch to the remote (e.g., GitHub) and sets tracking.  
Useful for syncing local work to remote.**

**git branch -d <branch-name>**

**Use:  
Deletes the specified local branch.  
Only works if the branch is fully merged or not in use.**

**git branch -a**

**Use:  
Lists all branches, both local and remote.  
Helpful for seeing what exists on GitHub and locally.**

**git push origin --delete <branch-name>**

**Use:  
Deletes a remote branch from GitHub or another remote.  
Example: git push origin --delete branch3.**

**git branch <branch-name>**

**Use:  
Creates a new branch without switching to it.  
Example: git branch branch2 creates the branch branch2.**

