# Manish Kshettri

# Git & GitHub Practical Lab Report

## Lab 1: Git Installation & Setup

* What is the command to check if Git is installed on your computer?

git --version

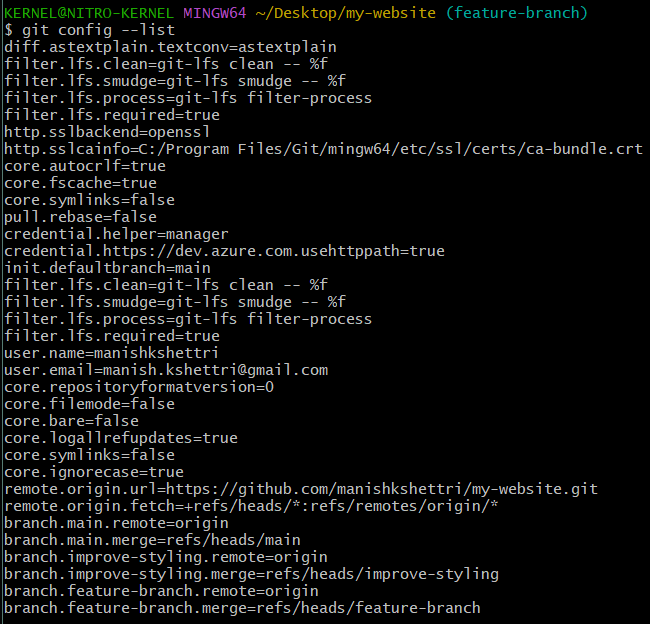
* Install Git on your computer using git-scm.com.
* Configure your Git username and email. Write the commands you used.

git config --global user.name "manishkshettri"  
git config --global user.email "manish.kshettri@gmail.com"

* What is the importance of setting your username and email in Git?

It identifies who made the changes in a project. Every commit includes the author’s name and email, useful for tracking and collaboration.

* Run git config --list and paste a screenshot of your configuration output.



## Lab 2: Create a Local Website Project

* Create a folder named my-website on your Desktop.

cd Desktop  
mkdir my-website  
cd my-website

* Inside the folder, create two files: index.html and style.css.
* Write a simple HTML code that displays a heading: “Welcome to My Website”.

<!DOCTYPE html>

<html>

<head>

<title>My Website</title>

<link rel="stylesheet" href="style.css">

</head>

<body>

<h1>Welcome to My Website</h1>

<p>This is my first Git project! < /p>

</body>

</html>

* Add CSS to make the heading text blue.

body {

font-family: Arial, sans-serif;

margin: 40px;

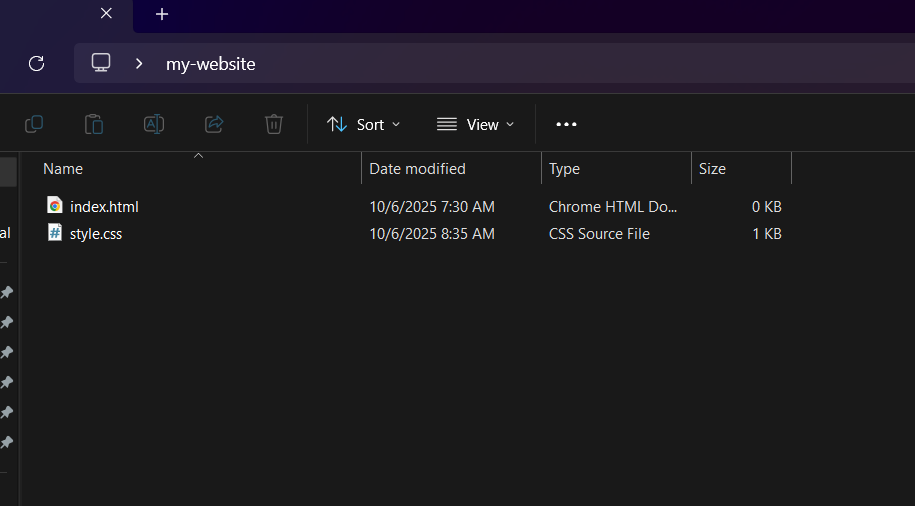
}

h1 {

color: blue;

}

* Show a screenshot of your folder structure and file contents.



## Lab 3: Initialize Git Repository

* What command do you use to navigate to your project folder in Command Prompt?

cd Desktop/my-website

* Initialize Git inside the my-website folder. Write the exact command used.

git init

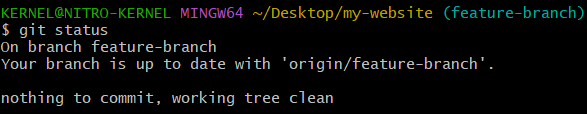
* What does the .git folder represent?

It’s a hidden folder that stores all version history, configurations, and commits.

* Run git status. What information does it display?

It shows current branch, untracked or staged files, and suggestions for next actions.

* Paste a screenshot showing successful initialization and Git status output.



## Lab 4: Add and Commit Files

* What is the purpose of the staging area in Git?

It’s where files are prepared before committing.

* Add index.html to the staging area. Write the command used.

git add index.html

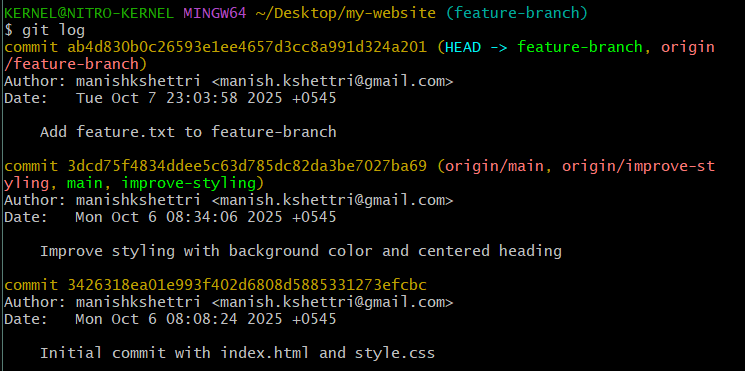
* Commit the HTML file with a message explaining your change.

git commit -m "Add index.html with welcome heading"

* Repeat for style.css.

git add style.css  
git commit -m "Add style.css with blue heading style"

* Run git log and note down the commit IDs and messages.



## Lab 5: Create GitHub Repository and Push Code

* Create a GitHub account (if not already created).

https://github.com

* Create a new repository named my-website (public, no README).
* Write the command to connect your local repo with your GitHub repo.

git remote add origin https://github.com/manishkshettri/my-website.git

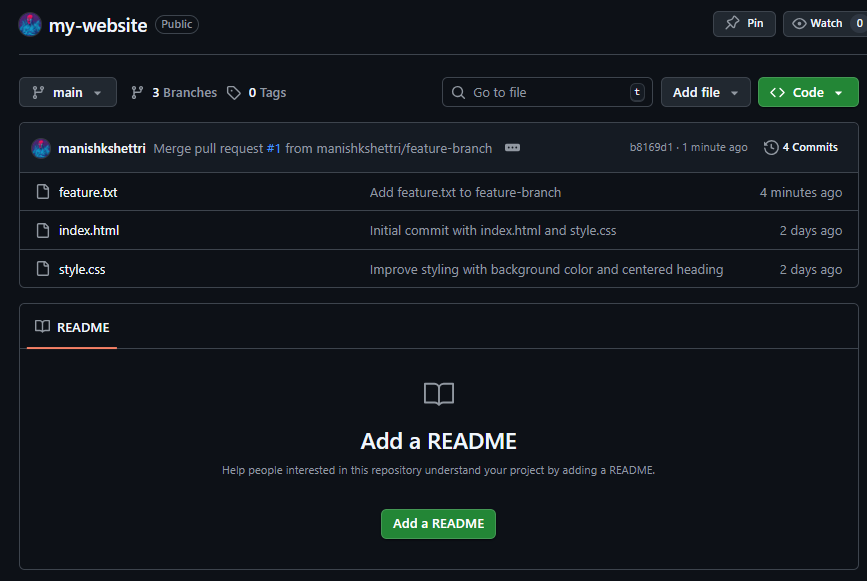
* Push your commits to GitHub.

git branch -M main  
git push -u origin main

* What does the command git push -u origin master do?

It pushes commits from your local master branch to the remote origin on GitHub and sets it as the upstream branch.

* Paste a screenshot of your GitHub repository showing both files.



## Lab 6: Create and Switch Branches

* What is a branch in Git?

A branch is a separate line of development.

* Create a new branch named improve-styling.

git checkout -b improve-styling

* Write the command used to view all branches in the repository.

git branch

* Modify your style.css file by adding a background color and center-aligned heading.

body {  
 font-family: Arial, sans-serif;  
 margin: 40px;  
 background-color: #f0f0f0;  
}  
  
h1 {  
 color: blue;  
 text-align: center;  
}  
  
p {  
 color: #333;  
 line-height: 1.6;  
}

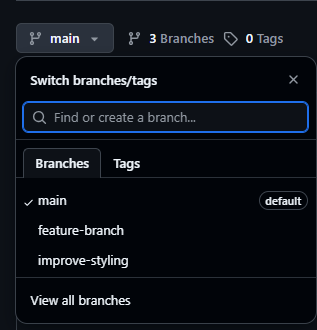
* Add and commit your updated file with a descriptive message.

git add style.css  
git commit -m "Improve styling with background color and centered heading"

* Push the branch to GitHub.

git push -u origin improve-styling

* Paste a screenshot showing both branches (master and improve-styling).



## Lab 7: Merge Branches

* What is the purpose of merging in Git?

To combine changes from one branch into another.

* Switch back to the master branch. Write the command used.

git checkout main

* Merge the improve-styling branch into master.

git merge improve-styling

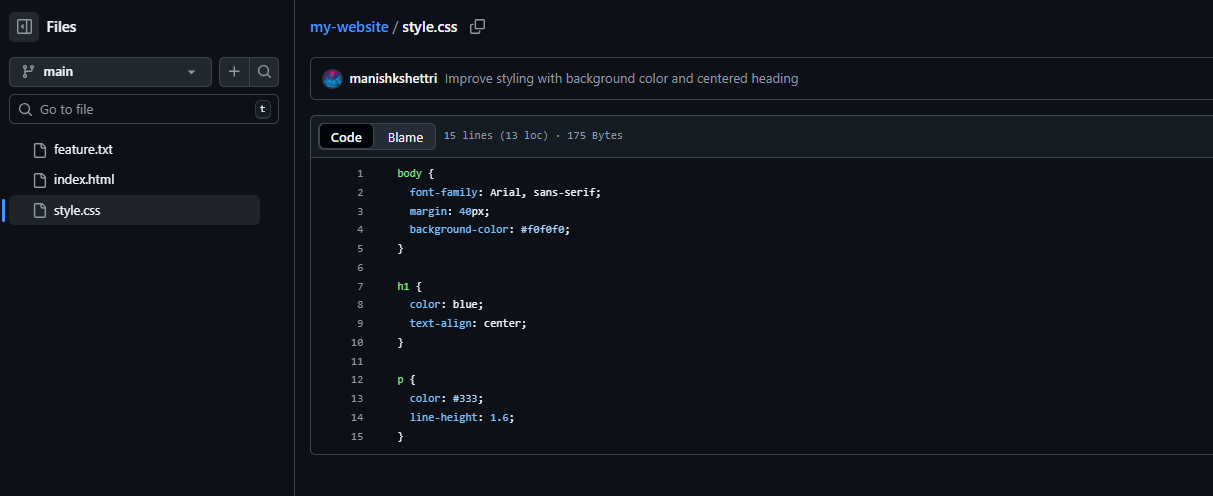
* Push the updated master branch to GitHub.

git push origin main

* What happens to the feature branch after merging?

It remains but can be safely deleted.

* Paste a screenshot showing the updated style.css file in the master branch.



## Lab 8: Collaboration Simulation

* What does it mean to 'fork' a repository on GitHub?

It creates a copy of someone else’s repository under your account.

* Fork a classmate’s repository to your own GitHub account.

Click Fork on their repo.

* Clone your forked repository to your computer.

git clone https://github.com/<your-username>/<repo-name>.git

* Create a new branch named add-footer.

git checkout -b add-footer

* Add a footer section to index.html and commit your changes.

<footer>  
 <p>© 2025 My Website. All rights reserved.</p>  
</footer>  
  
git add index.html  
git commit -m "Add footer section to index.html"

* Push your branch and create a Pull Request to the original repository.

git push -u origin add-footer

## Lab 9: Reflection Task

* In your own words, explain what Git is and why it is important.

Git is a version control system that tracks changes in code, supports collaboration, and manages versions.

* Explain the difference between git add, git commit, git push.

git add: Move files to staging area  
git commit: Save staged changes to local repository  
git push: Upload commits to remote repository

* What is a branch? Why do developers use it?

A branch is an independent workspace to develop new features without affecting main project.

* What command would you use to see the commit history?

git log

* What was the most challenging part of this lab series for you?

Understanding branching and merging initially.

* What skill did you gain that will help in your future coding projects?

Version control, GitHub collaboration, and professional project management.