Git & GitHub Practical Lab Questions

Lab 1: Git Installation & Setup

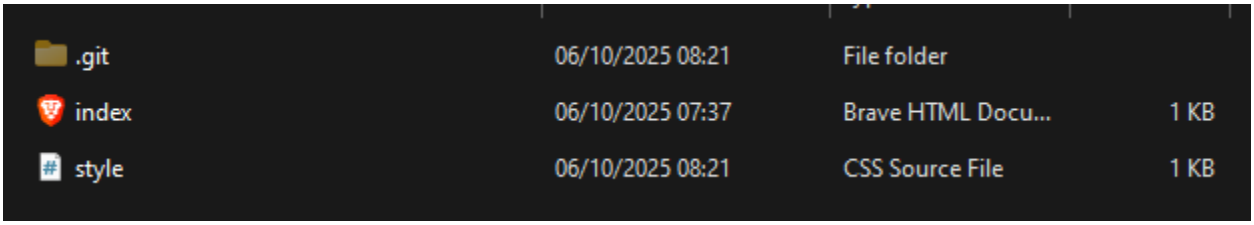
1. The command to check if Git is installed on a computer is “git –version”.
2. Installed Git.
3. The commands I used were:
    a. `git config --global user.name "Subarna Bajracharya"`
    b. `git config --global user.email buzzraws@gmail.com`
4. Setting username and email in Git my commits are properly identified and attributed to me.
5. Screenshot of configuration output is pasted below:

```
PS C:\Users\buzzr\Desktop\C\Online Class\Oct 06\my-website> git config --list
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=openssl
http.sslcainfo=C:/Program Files/Git/mingw64/etc/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
core.editor="C:\Users\buzzr\AppData\Local\Programs\Microsoft VS Code\bin\code" --wait
user.email=65555875+ItsBajra@users.noreply.github.com
user.name=Subarna Bajracharya
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
core.repositoryformatversion=0
core.filemode=false
core.bare=false
core.logallrefupdates=true
core.symlinks=false
```

Lab 2: Create a Local Website Project

Screenshot of folder structure and file contents:

Folder Structure:



Index.html:

```html
<!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>
```

Style.css:

```css
body {
    font-family: Arial, sans-serif;
    margin: 40px;
    background-color: #f0f0f0;
}

h1 {
    color: blue;
    text-align: center;
}

p {
    color: #333;
    line-height: 1.6;
}
```

Lab 3: Initialize Git Repository

1. We use ‘cd’ command to navigate to project folder in command prompt. Specifically, it goes this way: cd path\to\your\project\folder.

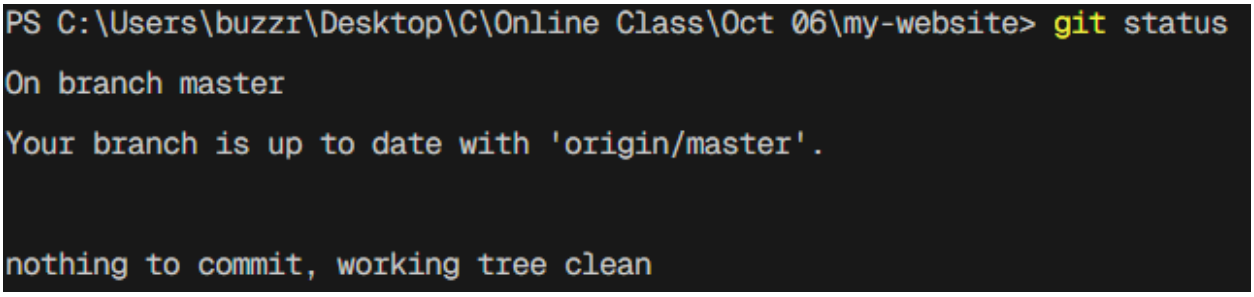
2. We used to ‘cd’ command to first navigate to the project folder and then we used the ‘git init’ command to initialize git inside the folder.

3. The .git folder stores all the information and history for my git repository.

4. When running the ‘git status’ command, it displays the following information:
   On branch master
   Your branch is up to date with 'origin/master'.

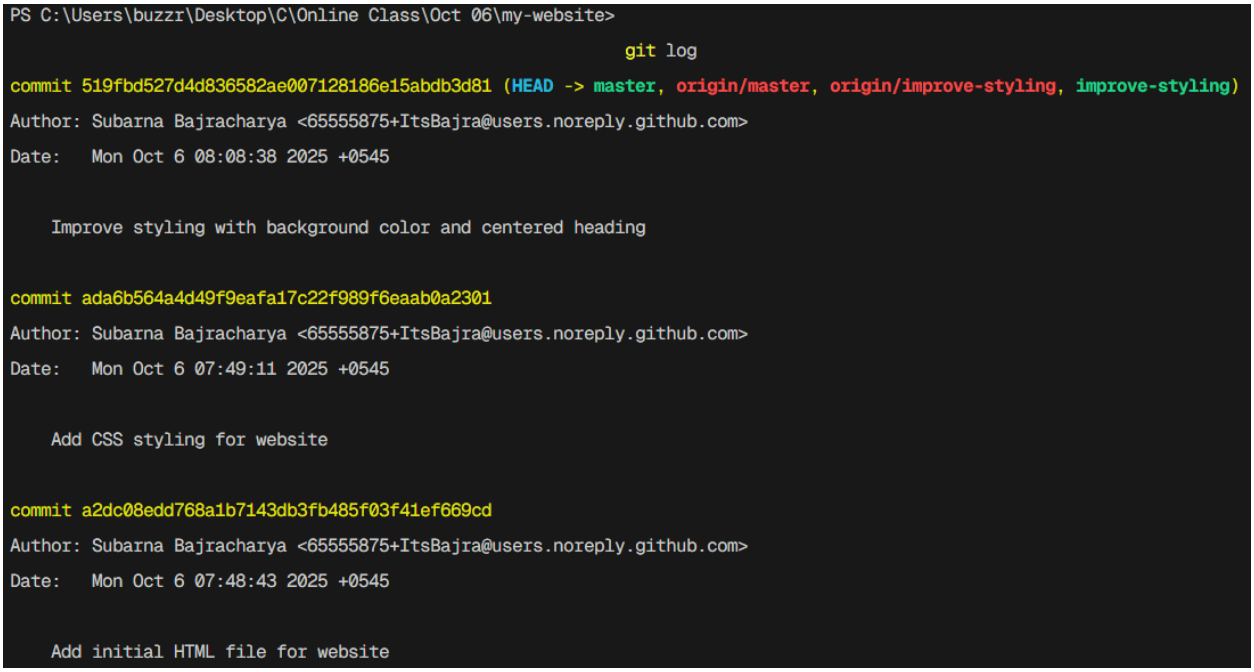
   nothing to commit, working tree clean

5. Screenshot of git status output :



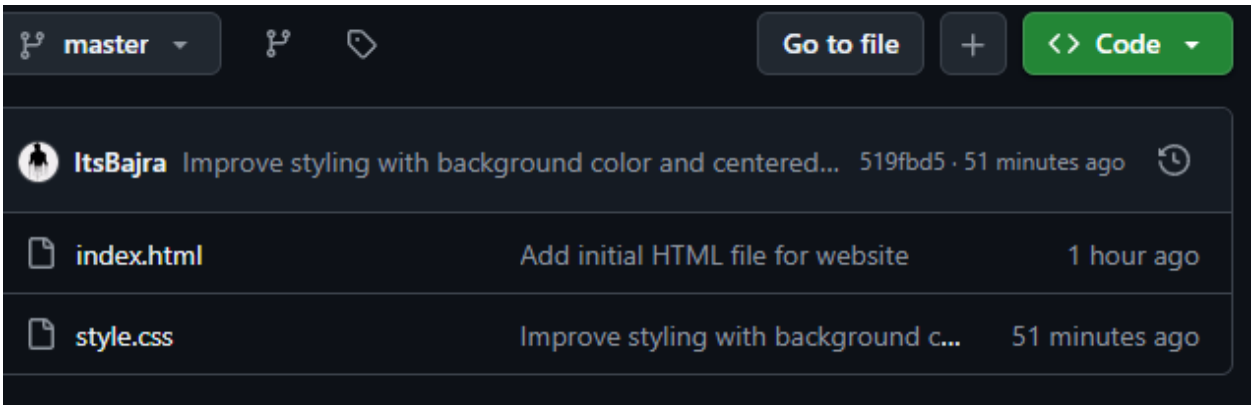
Lab 4: Add and Commit Files

1. The staging area in git allows us prepare and review changes before committing them to the repository.
2. The command used to add index.html to the staging area was: ‘git add index.html’
3. Done
4. Done
5. Screenshot of git log output:



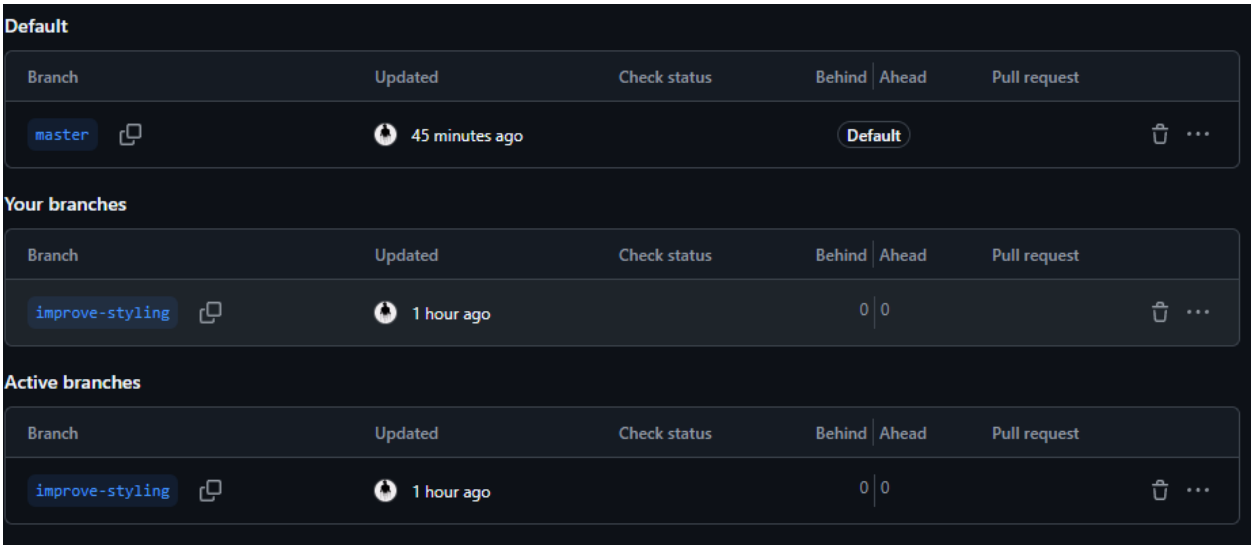
Lab 5: Create GitHub Repository and Push Code

A screenshot showing my GitHub Repository:



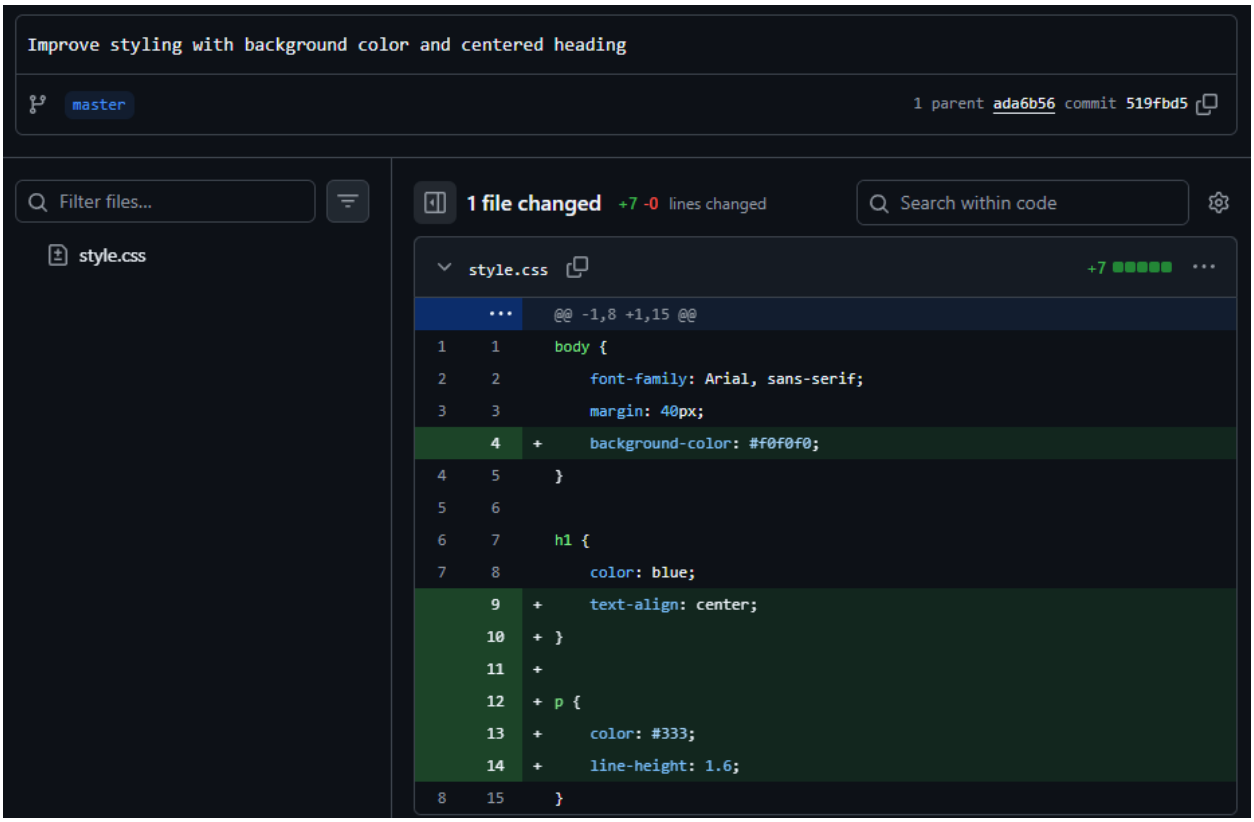
Lab 6: Create and Switch Branches

1. A branch is a separate line of development that allows us to work on new changes without affecting the main codebase.
2. Done
3. The command used to view all branches in the repository is ‘git branch’
4. Done
5. Done
6. Done
7. The screenshot of both branches:



Lab 7: Merge Branches

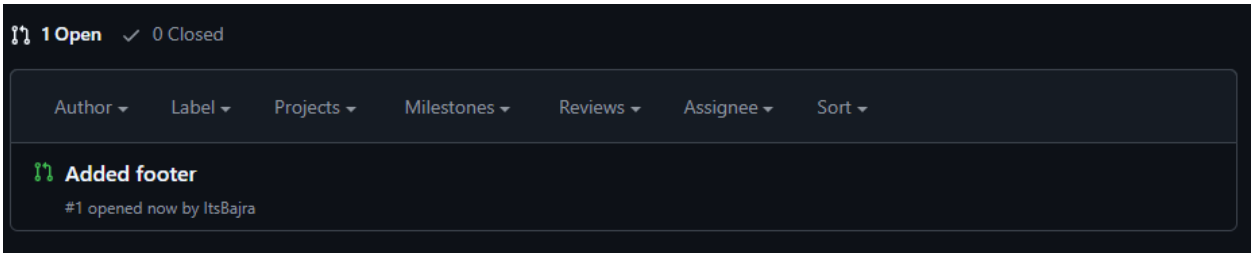
1. The purpose of merging git is to combine the changes from different branches into one single branch.
2. The command used to switch back to the master branch is ‘git checkout master’
3. Done.
4. Done.
5. After merging, the feature branch is still there in the repository and can be deleted if no longer needed.
6. Screenshot showing the updated style.css file in the master branch:



Lab 8: Collaboration Simulation

1. To fork a repository refers to creating a copy of someone else's repository under my account.

Screenshot of Pull Request Page:



Lab 9: Reflection Task

1. Git is a tool that helps us track changes on our code, collaborate with other people, and manage the versions of our project. This makes teamwork and code management much easier and safer.
2. The command git add is used to add files to staging. The command git commit is used to save the staged changes to local repo with a message and the command git push is used to push the saved changes from local repo to a remote repository.
3. A branch is a separate line of development that allows developers to work on new changes without affecting the main codebase.
4. The command 'git log' is used to see the commit history.
5. There wasn't any challenging part in this lab series as I'm quite familiar with git and GitHub.
6. I learned how to fork repositories, which will help me collaborate more effectively and contribute to other coding projects in the future.

Answers to the Git Quiz done in class:

Question 1: B

Question 2: A

Question 3: C

Question 4: D

Question 5: D

Question 6: B