Department of
Computer Science and Engineering

Title: Version Control (Using Git) Part II

Software Testing & Quality Assurance Lab CSE 434

Green University of Bangladesh

1 Objective(s)

- To understand the concepts of remote repositories in Git, particularly GitHub.
- To learn how to collaborate on projects using remote repositories.
- To grasp the fundamentals of branching in Git.

2 Introduction

In Part II of the Git lab, we will learn the utilization of remote repositories, specifically GitHub, and ex plore branching in Git. Understanding remote repositories and branching is crucial for collaborative software development and efficient project management.

3 Remote Repositories and GitHub

3.1 Understanding Remote Repositories

A remote repository is a Git repository hosted on a server, separate from your local machine. Remote repositories facilitate collaboration among multiple developers by providing a centralized location to share code and track changes.

3.2 Introduction to GitHub

GitHub is a popular platform for hosting Git repositories and collaborating on projects. It offers features such as issue tracking, pull requests, and project management tools, making it an invaluable resource for software development teams.

3.3 Basic Operations with Remote Repositories

- Cloning: To clone a remote repository to your local machine, use the command git clone <repository_URL>.
- Pushing Changes: To push your local changes to a remote repository, use git push origin <branch_name>. - Pulling Changes: To fetch and merge changes from a remote repository to your local repository, use git pull origin <branch_name>.

4 Github Remote Repository

Follow the steps below to setup github.
Login to github and create a new repository.

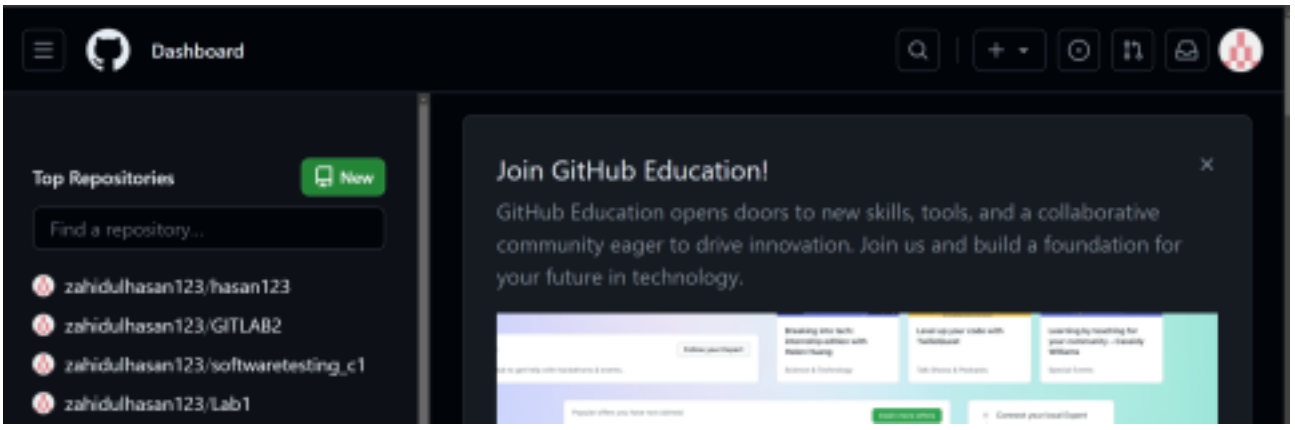

Figure 1: Github dashboard

Give a name for the project.

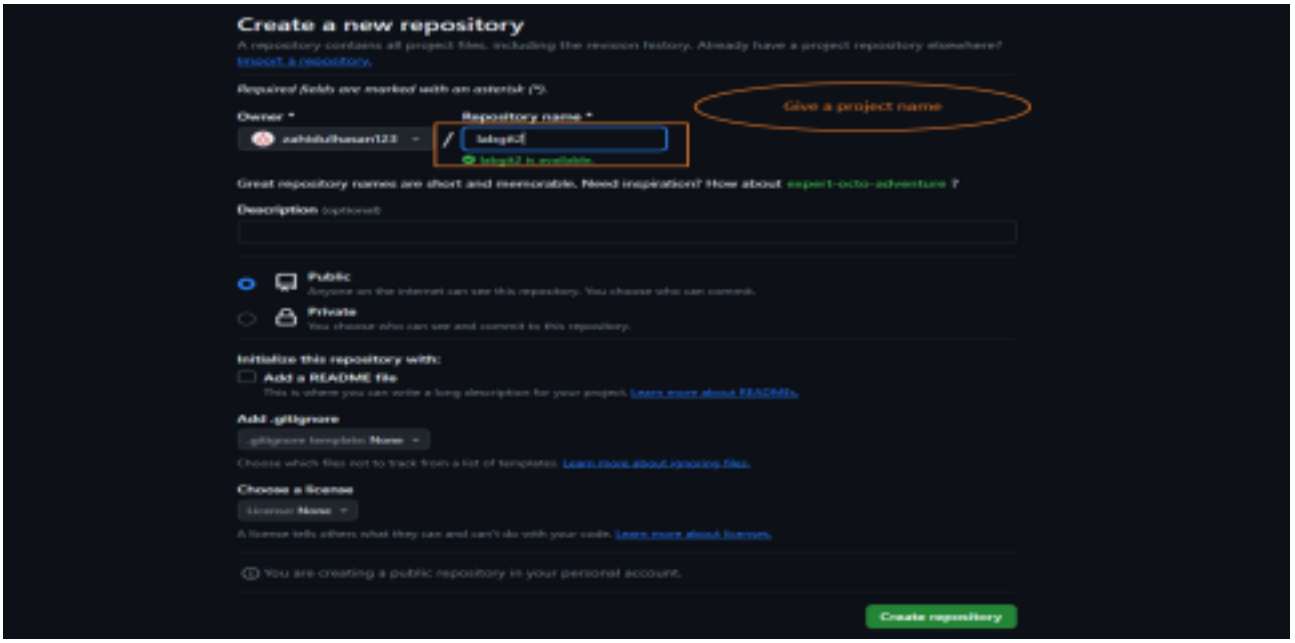

Figure 2: Creating a new repository

Copy all the texts as shown below:

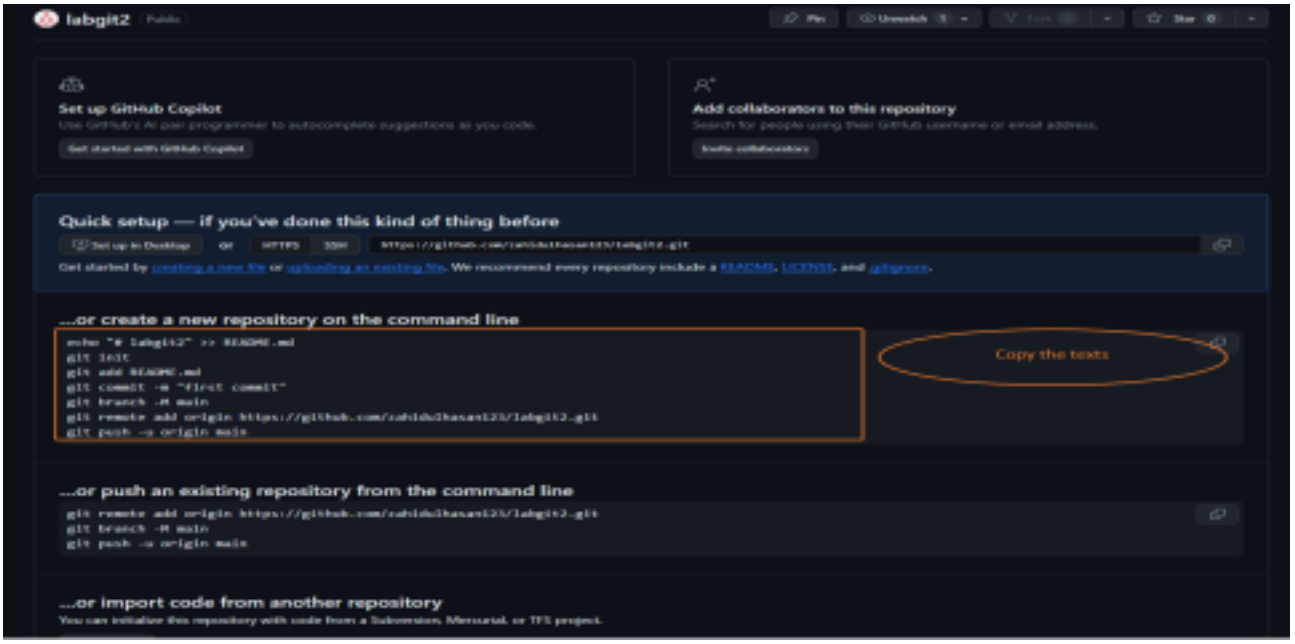

Figure 3:

Paste the text command in the terminal/ Git bash. Press enter.

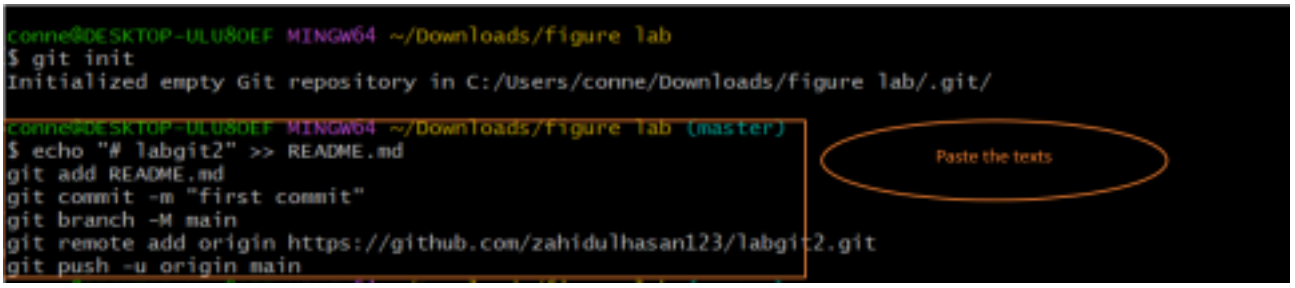

Figure 4:

Now the project will be pushed into the GitHub remote repository.

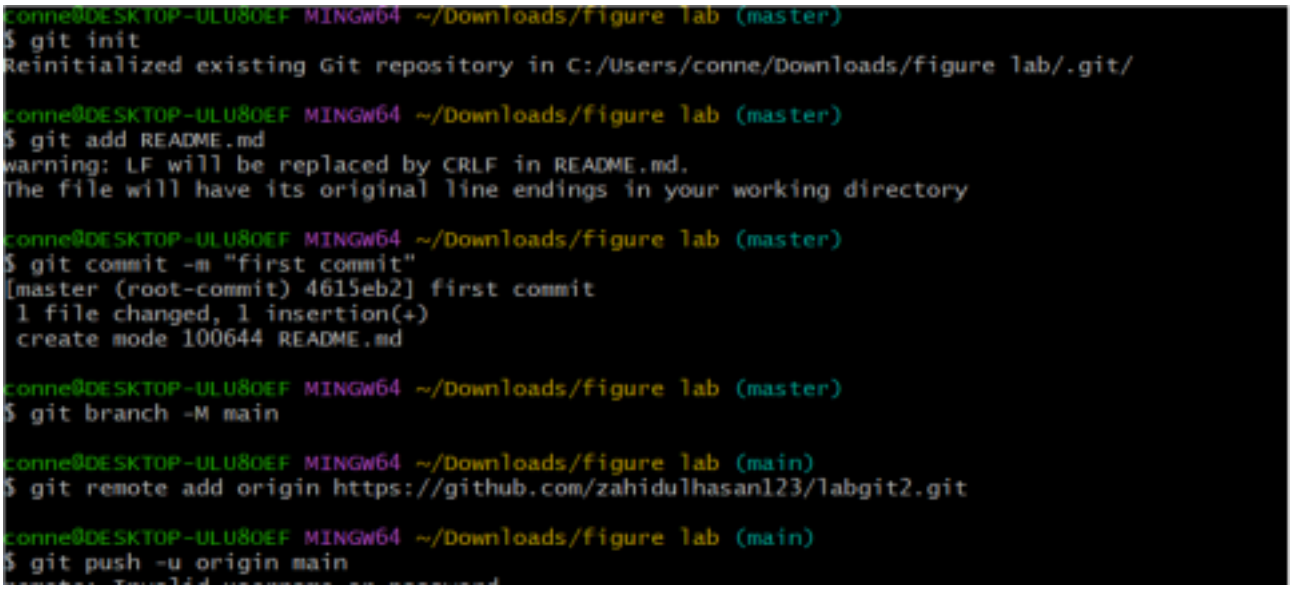

Figure 5:

This error may occur while pushing the project files because of wrong user and credentials:

Figure 6:

To solve the problem, firstly, remove all users using the below command:

Figure 7:

Setup your account using email and Github username

```
conne@DESKTOP-ULU8OEF MINGW64 ~/Downloads/figure lab (main)
$ git config --global user.name "zahidulhasan123"

conne@DESKTOP-ULU8OEF MINGW64 ~/Downloads/figure lab (main)
$ git config --global user.email "zahidul_hasan@cse.green.edu.bd"

conne@DESKTOP-ULU8OEF MINGW64 ~/Downloads/figure lab (main)
$ git config user.email
zahidul_hasan@cse.green.edu.bd

conne@DESKTOP-ULU8OEF MINGW64 ~/Downloads/figure lab (main)
$ git config user.name
zahidulhasan123
```

Figure 8:

Now push the files to the remote repository.

```
conne@DESKTOP-ULU8OEF MINGW64 ~/Downloads/figure lab (main)
$ git push -u origin main
```

Figure 9:

A popup will appear in windows OS. Give the access token here. It is discussed in the next steps how you can get an access token (In Linux, you will be asked to enter username and password. give username and then give access token in the password field. Keep in mind that, do not enter password, instead enter the access token).

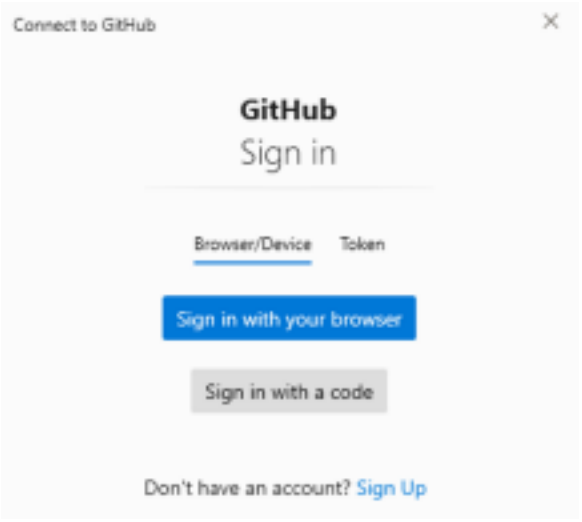

Figure 10:

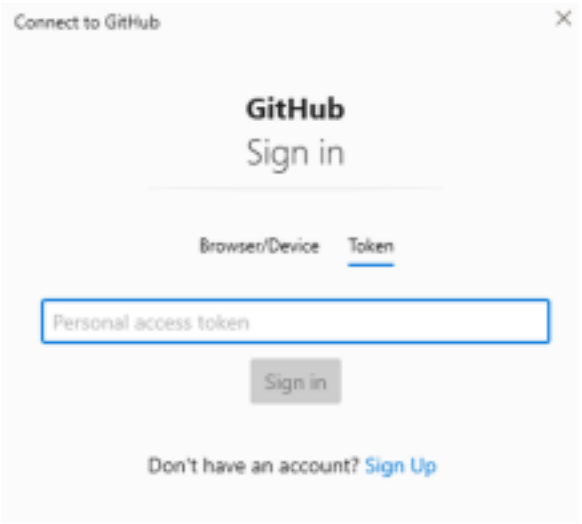

Figure 11:

Go to Github settings.

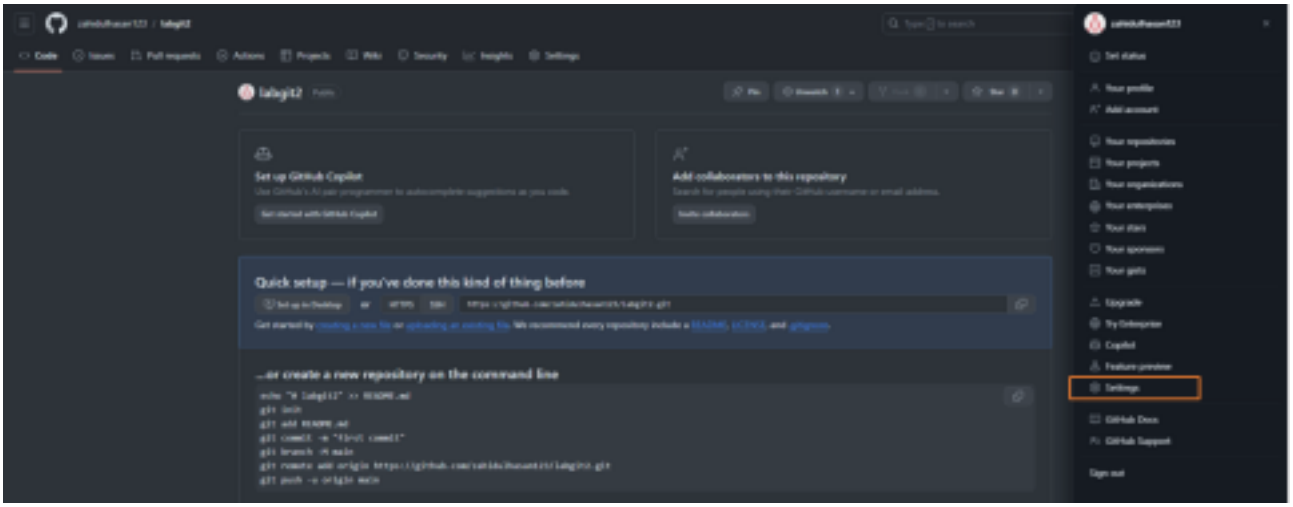

Figure 12:

Click on Developer settings.

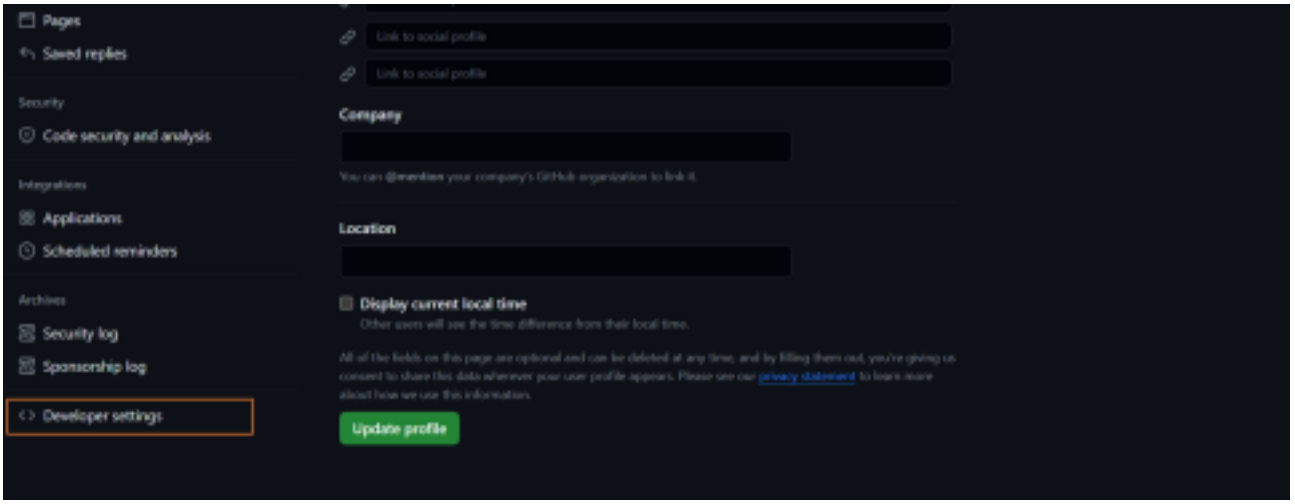

Figure 13:

Click on Personal access token -> Tokens (Classic).

Figure 14:

Click generate new token.

Figure 15:

Give a note for the token in input field.

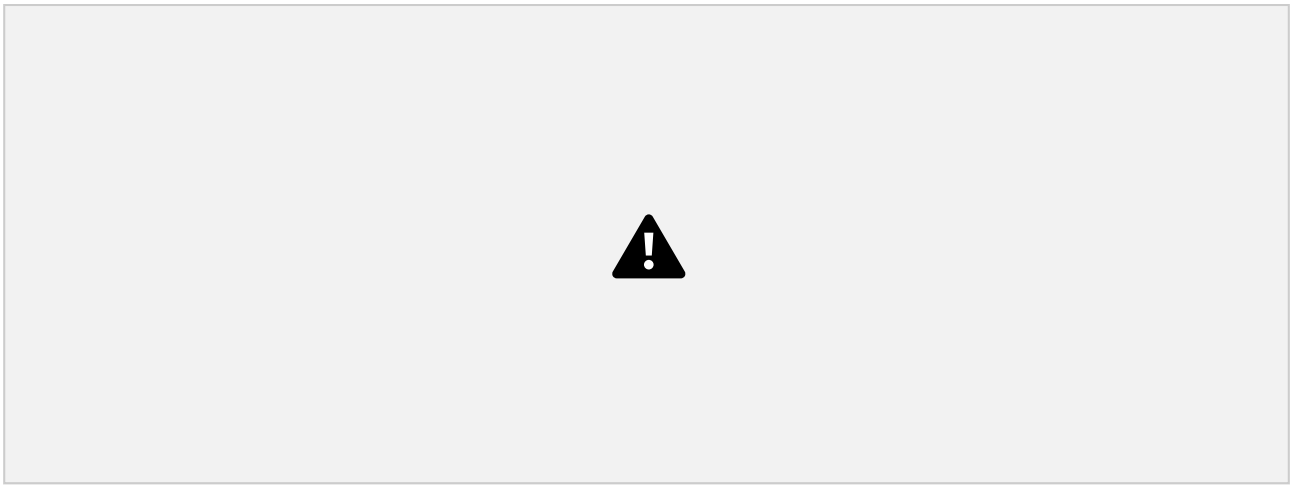

Figure 16:

Give necessary permissions and Generate token.

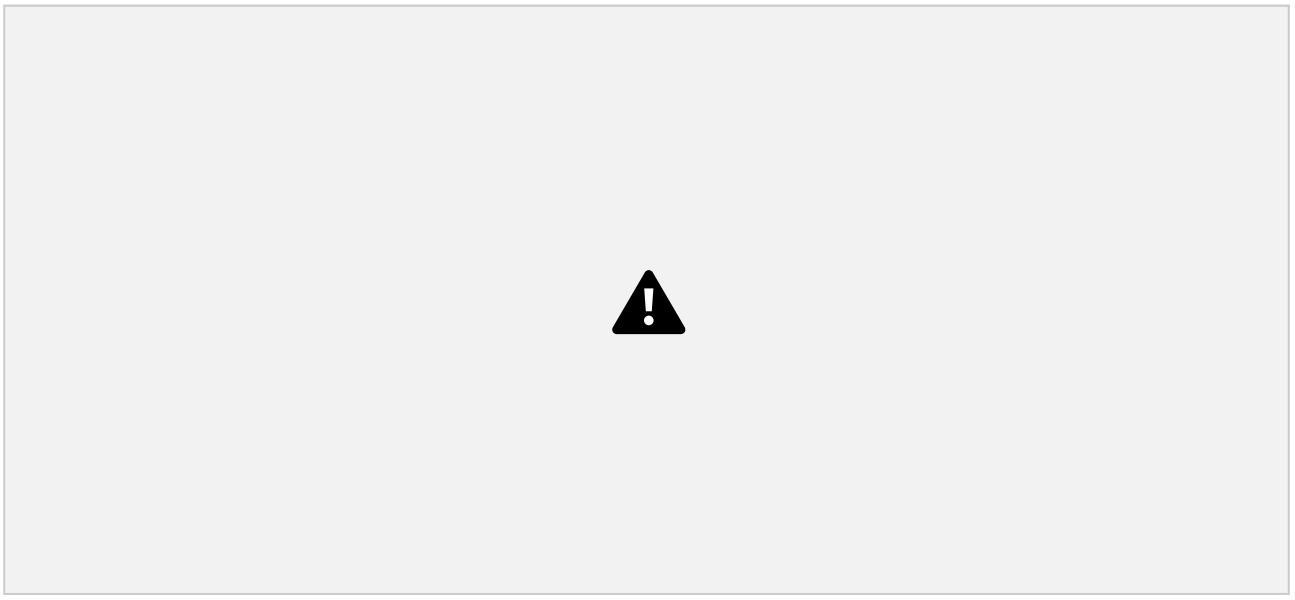

Figure 17:

Paste the token and click on Sign in.

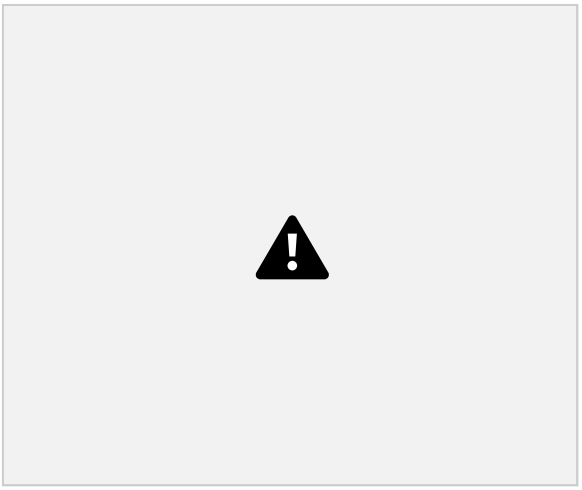

Figure 18:

Now, The files have been uploaded to the remote repository.

Figure 19:

This way make change in the project files and then add them to staging directory and then commit those changes. After committing, you can push the files to the main remote branch.

5 Understanding Branches

A branch in Git is a lightweight movable pointer to a commit. Branches allow developers to work on features or bug fixes independently without affecting the main codebase.

5.1 Basic Operations with Branches

- Creating a Branch: To create a new branch, use git branch <branch_name>.
- Switching Branches: To switch to a different branch, use git checkout <branch_name>. - Merging Branches: To merge changes from one branch into another, use git merge <branch_name>.

6 Lab Task (Please implement yourself and show the output to the instructor)

1. Create a new repository on GitHub.
2. Clone the repository to your local machine.
3. Create a new branch for a feature or bug fix.
4. Make changes to your project files.
5. Add and commit your changes to the branch.
6. Push the branch to the remote repository on GitHub.
7. Create a pull request on GitHub to merge your branch into the main branch.
8. Review and merge the pull request on GitHub.
9. Switch back to the main branch locally and pull the changes from the remote repository.

7 Lab Exercise (Submit as a report)

Create a document file and include the following in your submission:

1. The list of commands you used for completing the lab task.
2. Screenshots of relevant commands and outputs from your terminal or Git Bash.
3. A brief description of your experience completing the lab task, including any challenges you faced and how you overcame them.

8 Policy

Copying from the internet, classmates, seniors, or any other source is strictly prohibited. Any such copying will result in a deduction of 100% marks if detected.