**GIT LAB ASSIGNMENT -02**

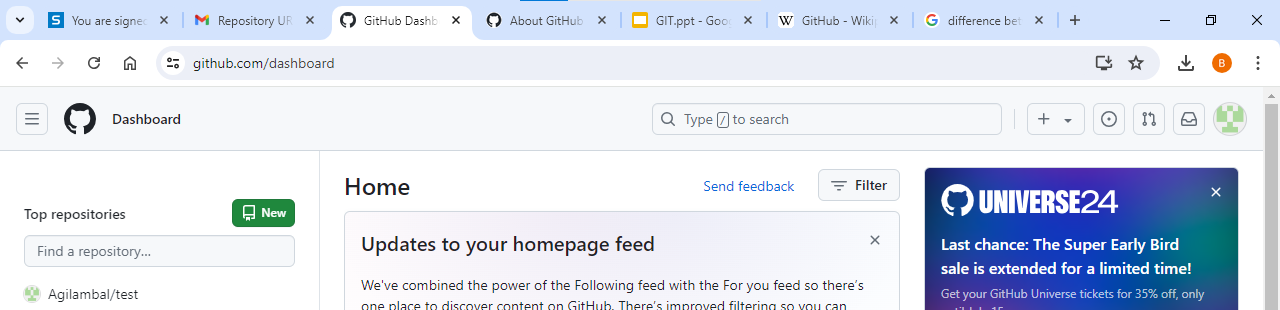
NAME: SYED ALI FATHIMA M

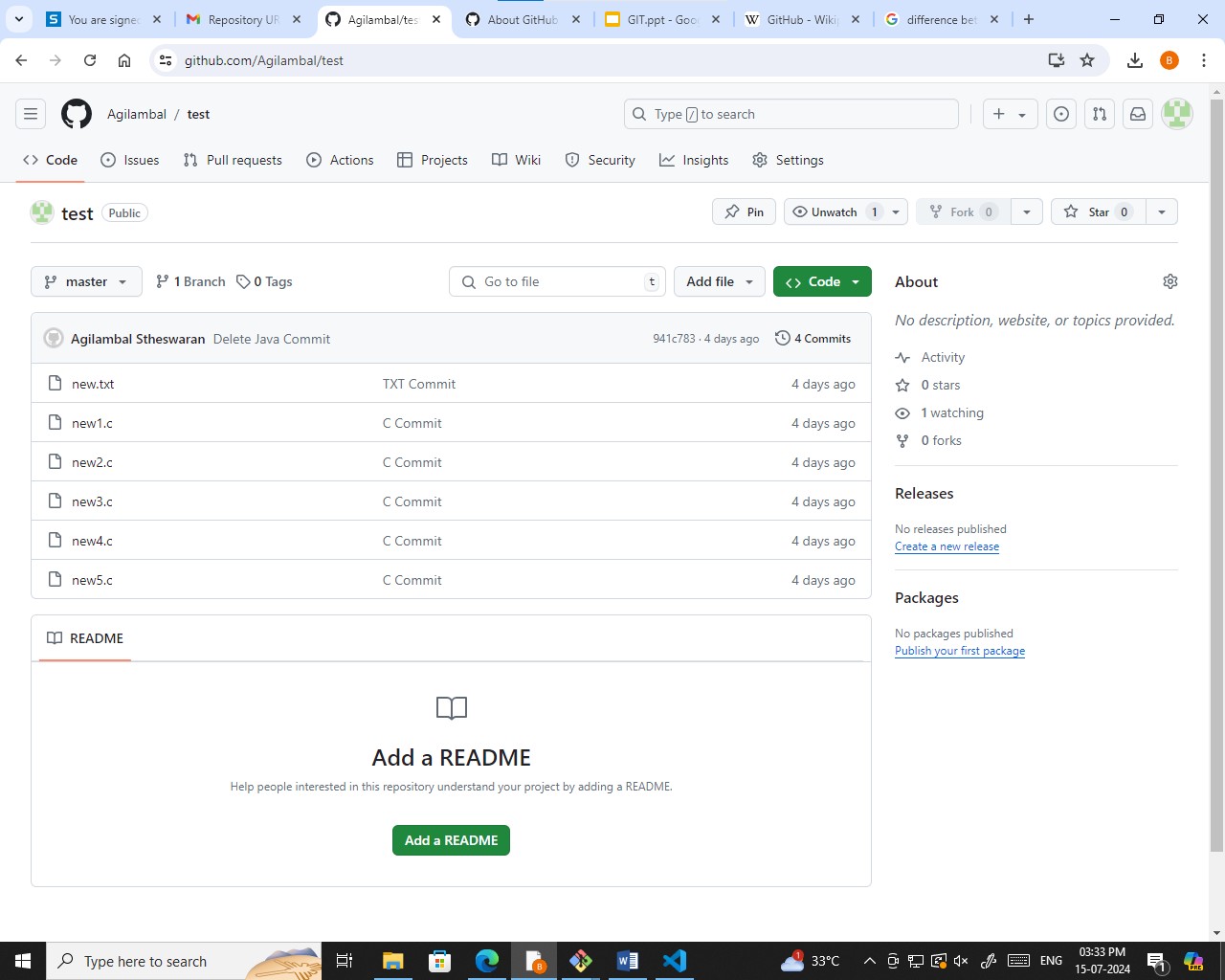
ROLL NO:731621121055

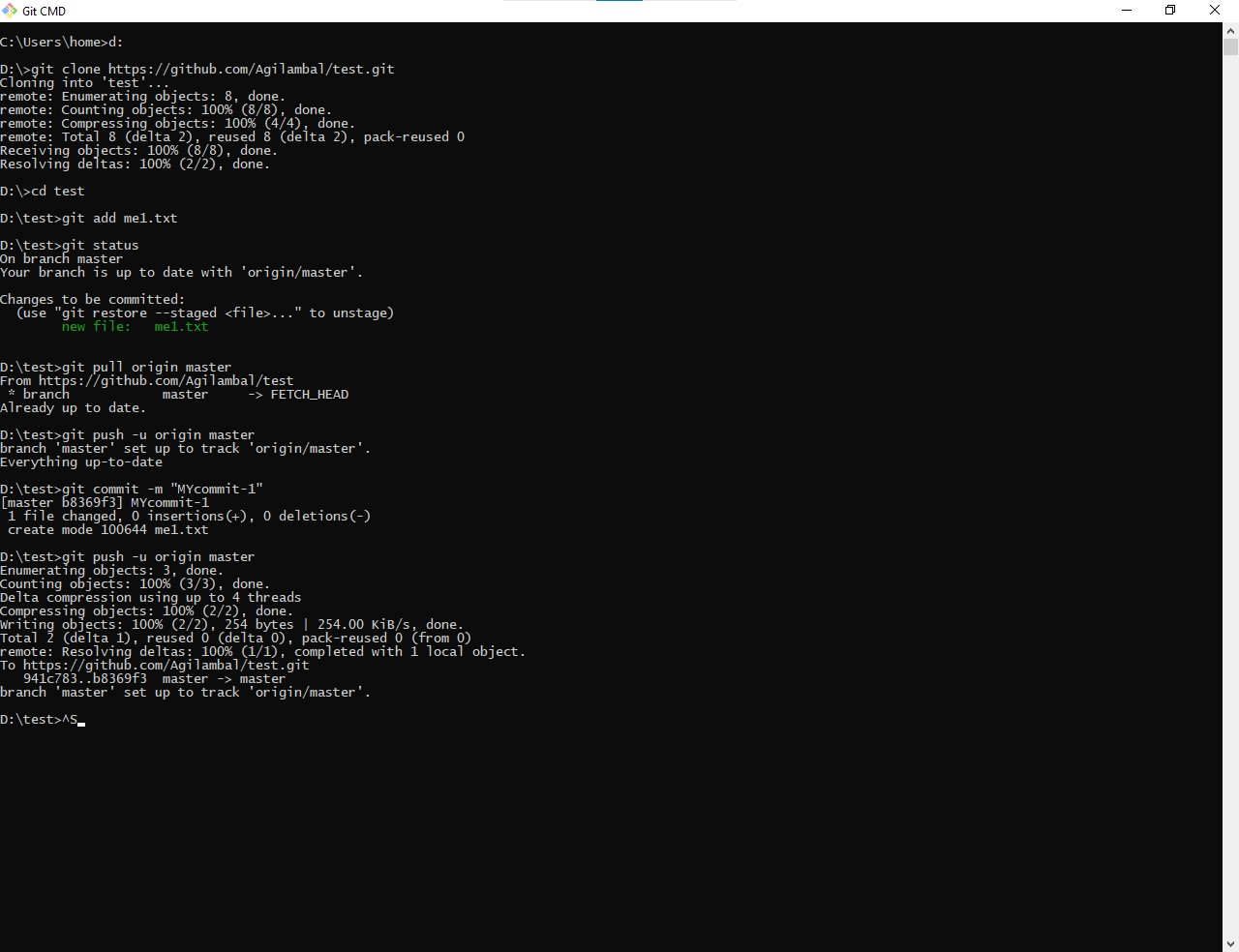
E-MAIL ID:fathimamusthakali@gmail.com

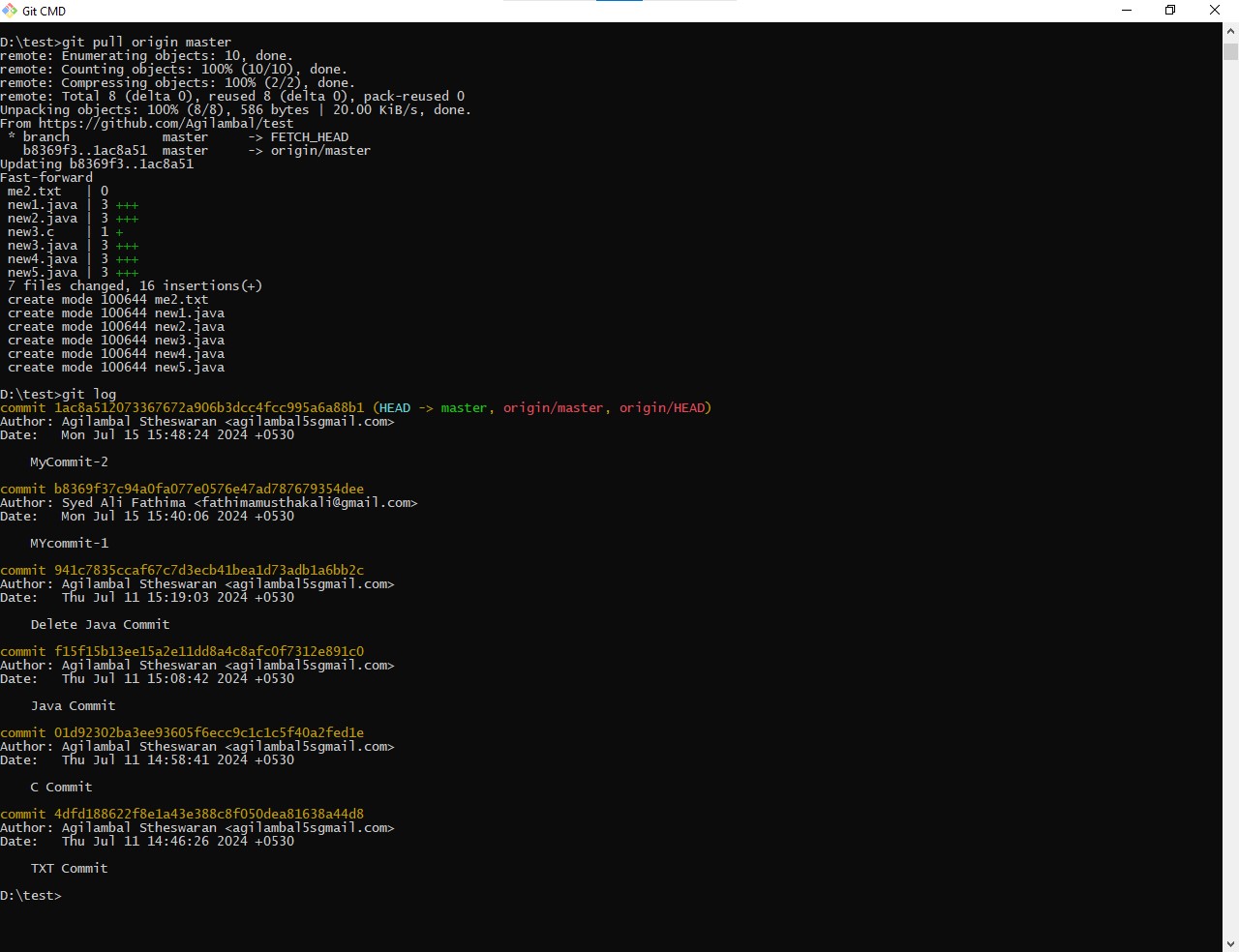
**EXERCISE – 3**

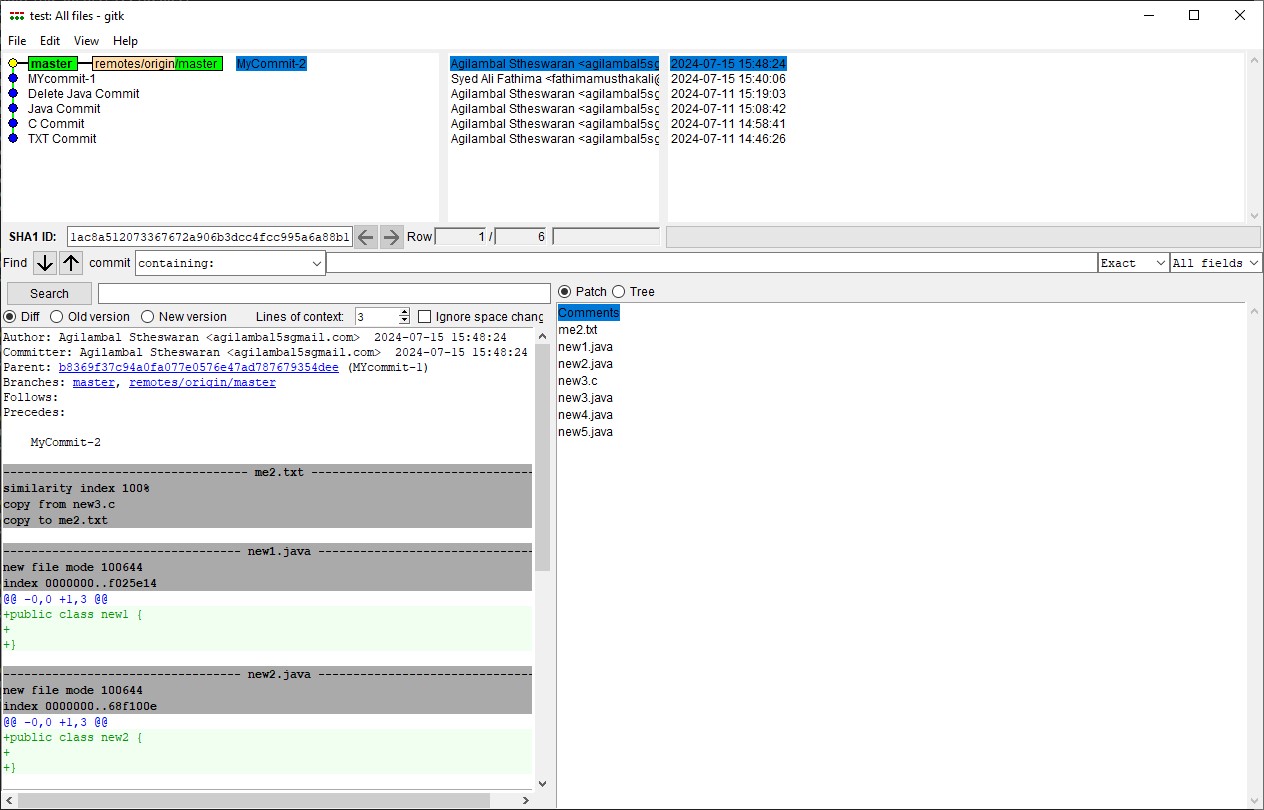
1. **First, one person in the group should create a public repository using their GitHub account.**

****

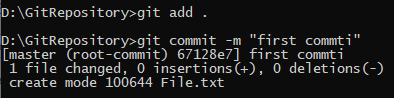
1. **This same person should then follow the instructions from GitHub to add a remote, and then push their repository. Do not forget the –u flag, as suggested by GitHub!**
2. **All of the other members of the group should then be added as collaborators, so they can commit to the repository also.**
3. **Next, everyone else in the group should clone the repository from GitHub. Verify that the context of the repository is what is expected.**
4. **One of the group members who just cloned should now make a local commit, then push it. Everyone should verify that when they pull, that commit is added to their local repository (use git log to check for it).**



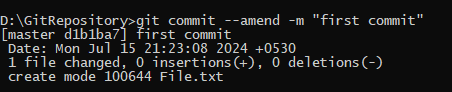
1. **Look at each other’s git log output. Notice how the SHA-1 is the same for a given commit across every copy of the repository. Why is this important?**
2. **Two members of the group should now make a commit locally, and race to push it. To keep things simple, be sure to edit different files. What happens to the runner-up**

**8.The runner-up should now pull. As a group, look at the output of the command. Additionally, look at the git log, and notice that there is a merge commit. You may also wish to view the DAG in gitk.**

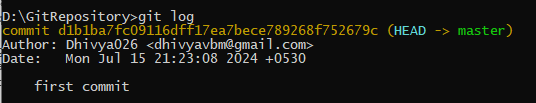
**EXERCISE – 4**

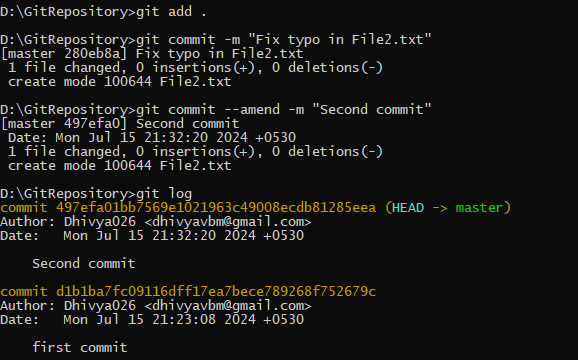
**1. Make a commit, and make a silly typo in the commit message.**

**2. Use the --amend flag to enable you to fix the commit message.**

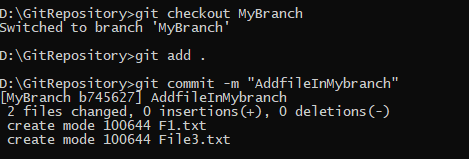


**3. Look at the log and notice how the mistake is magically gone.**

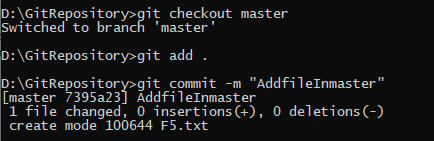


**4. Now make a commit where you make a typo in one of the files. Once again, use --amend to magic away your problems.**

**5. Create a branch. Make a commit.**



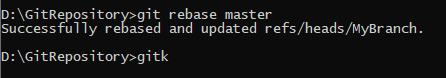
**6. Now switch back to your master branch. Make a (non-conflicting) commit there also.**



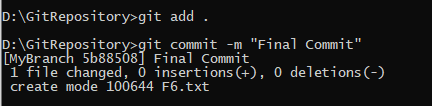
**7. Now switch back to your branch.**



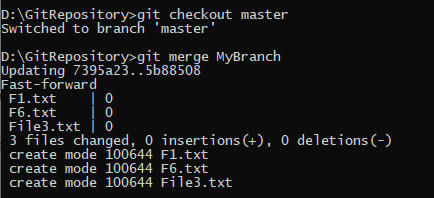
**8. Use the rebase command in your branch. Look at the DAG in gitk, and note that you have the commit from the master branch, but no merge commit.**



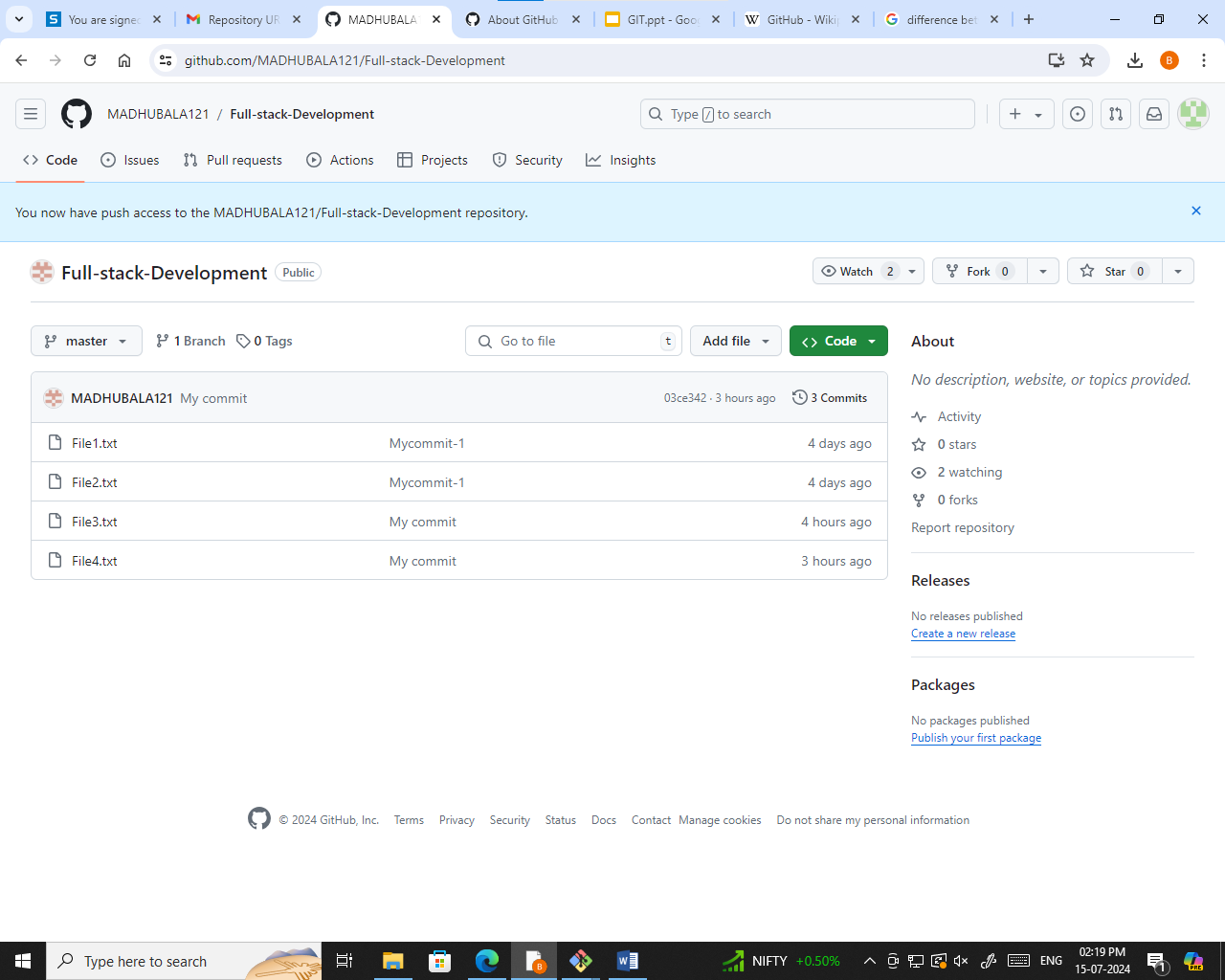
**9. Make one more commit in your branch.**



**10. Return to master. Merge your branch. Notice how, thanks to the rebase, this is a fast- forward merge.**



**3. All of the other members of the group should then be added as collaborators, so they can commit to the repository also.**

****