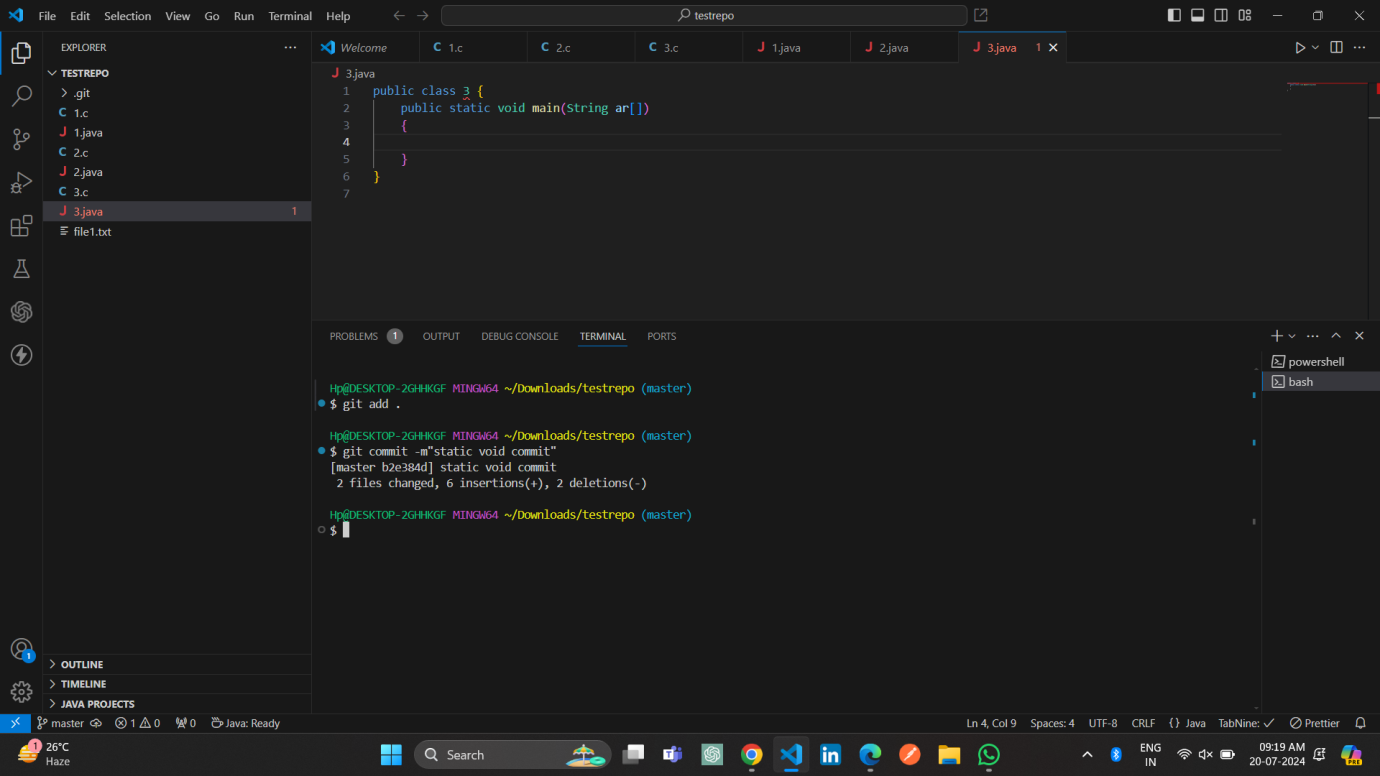
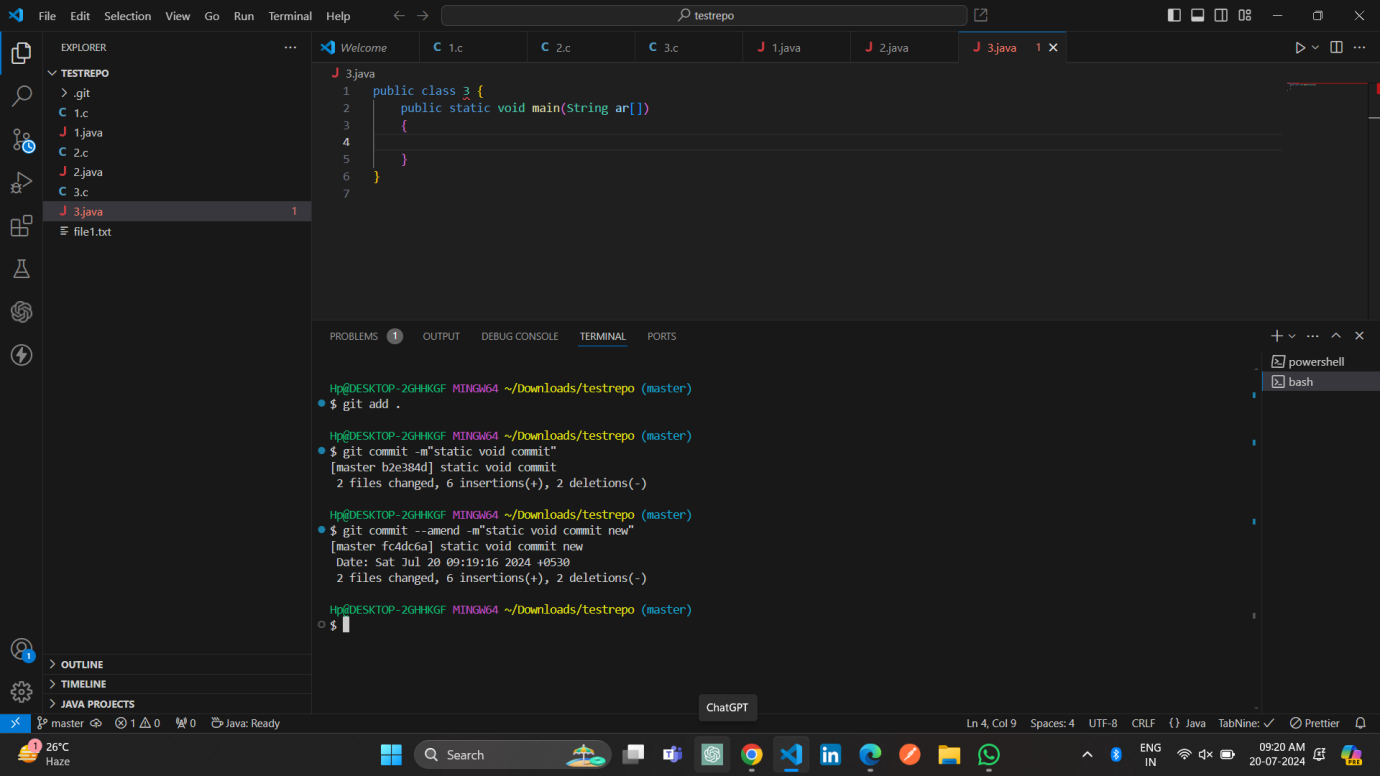
**Product Training Assignment 4 Ex1**

**Step 1:**

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

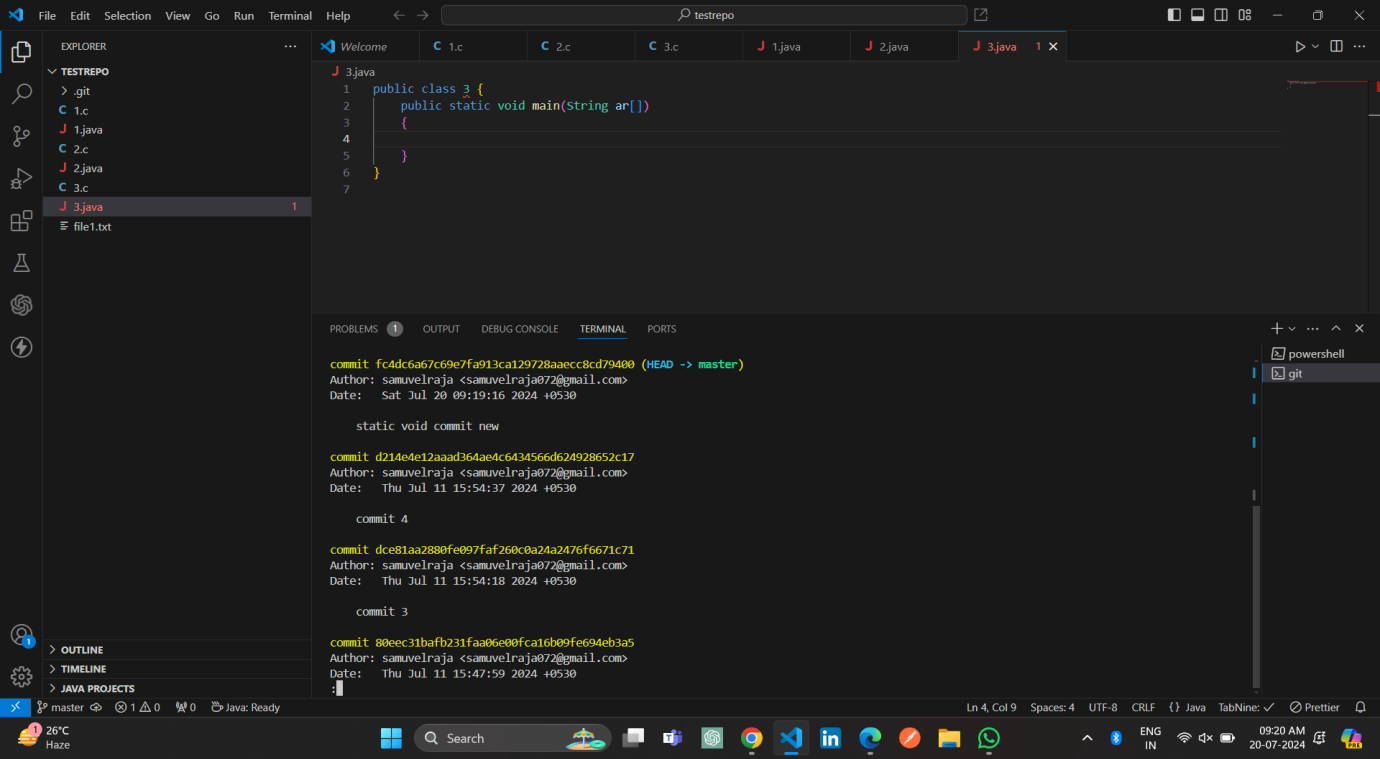
**Step 2:**

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

****

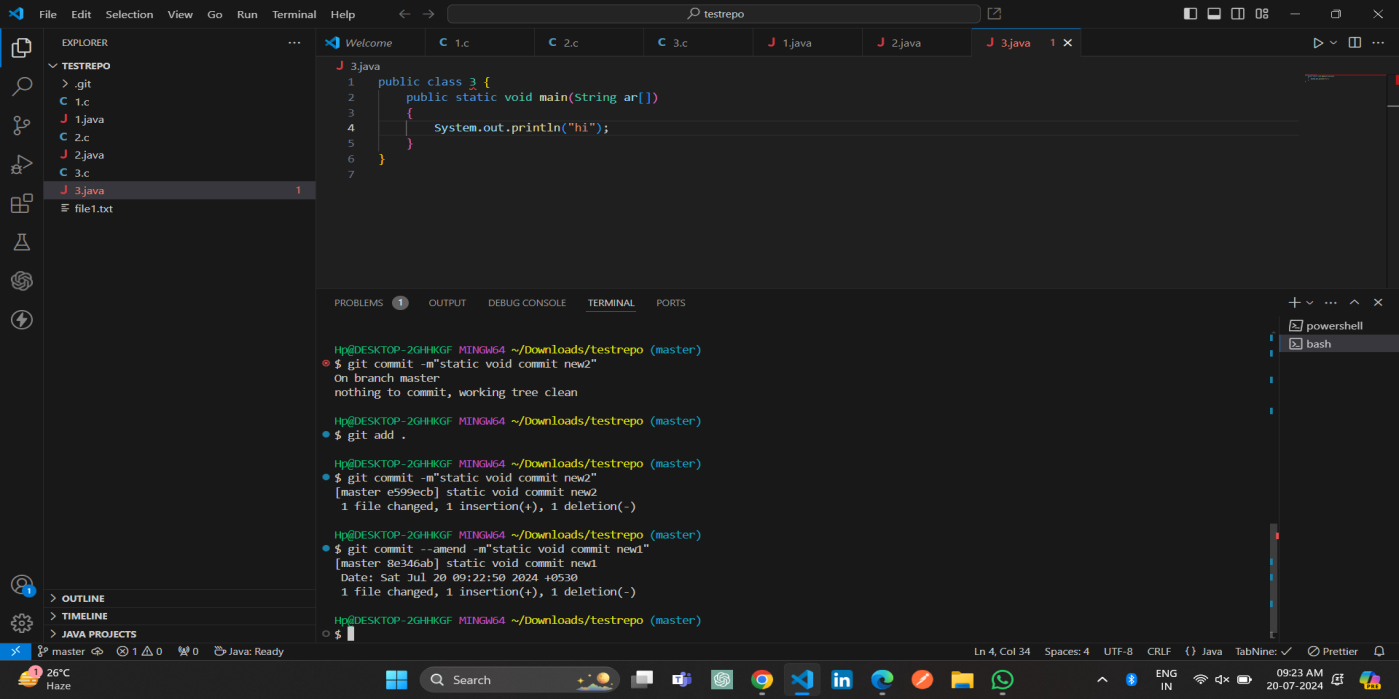
**Step 3:**

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

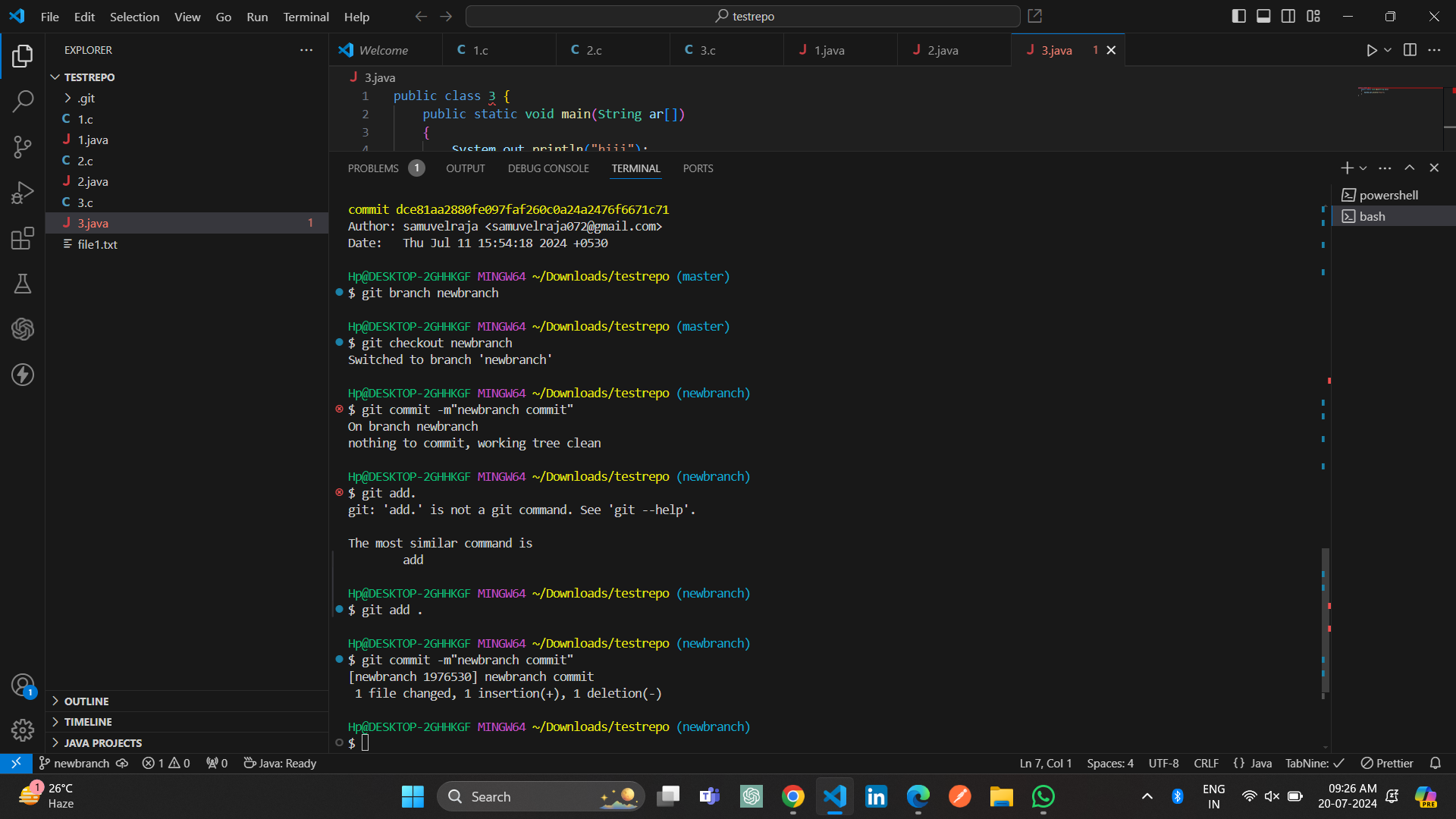


Step 4:

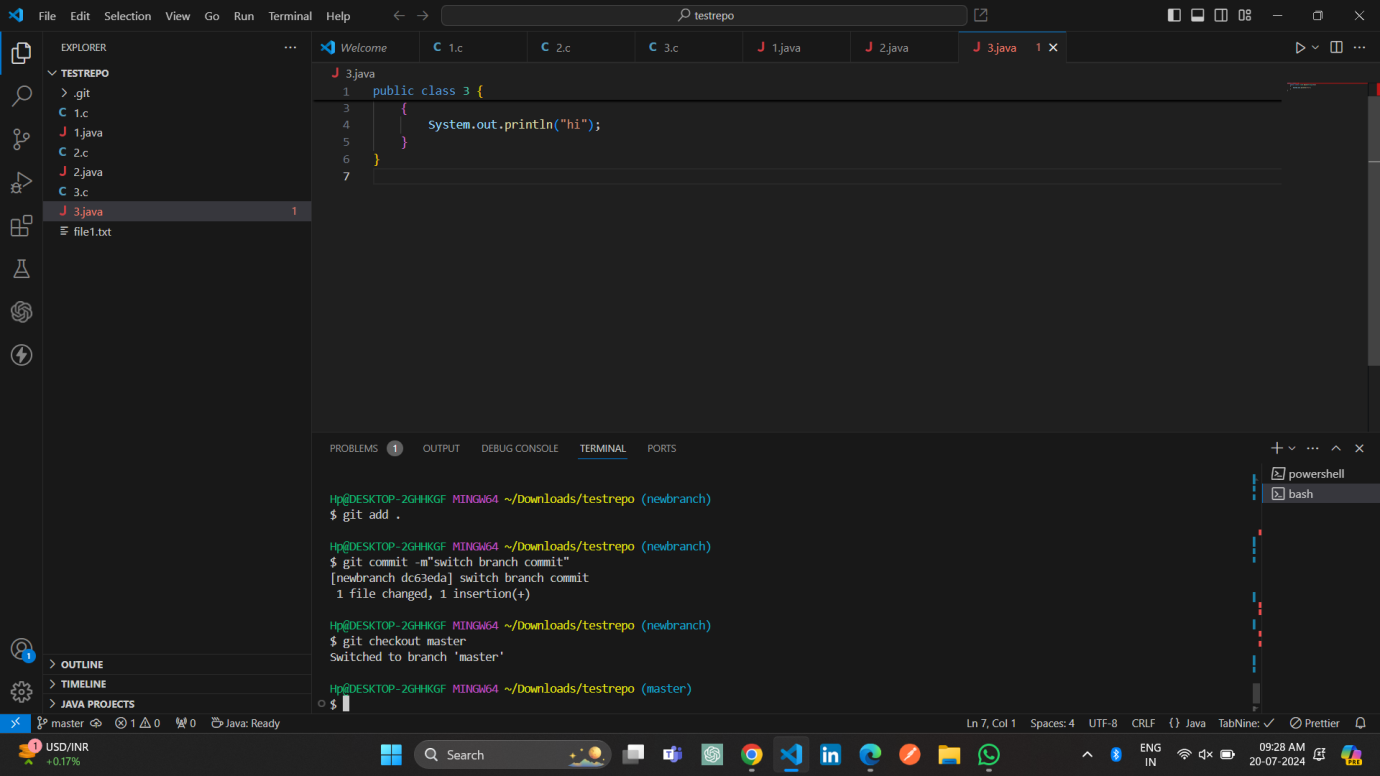
Now make a commit where you make a typo in one of the files. Once again, use --amend to magic away your problems.



**Step 5:**

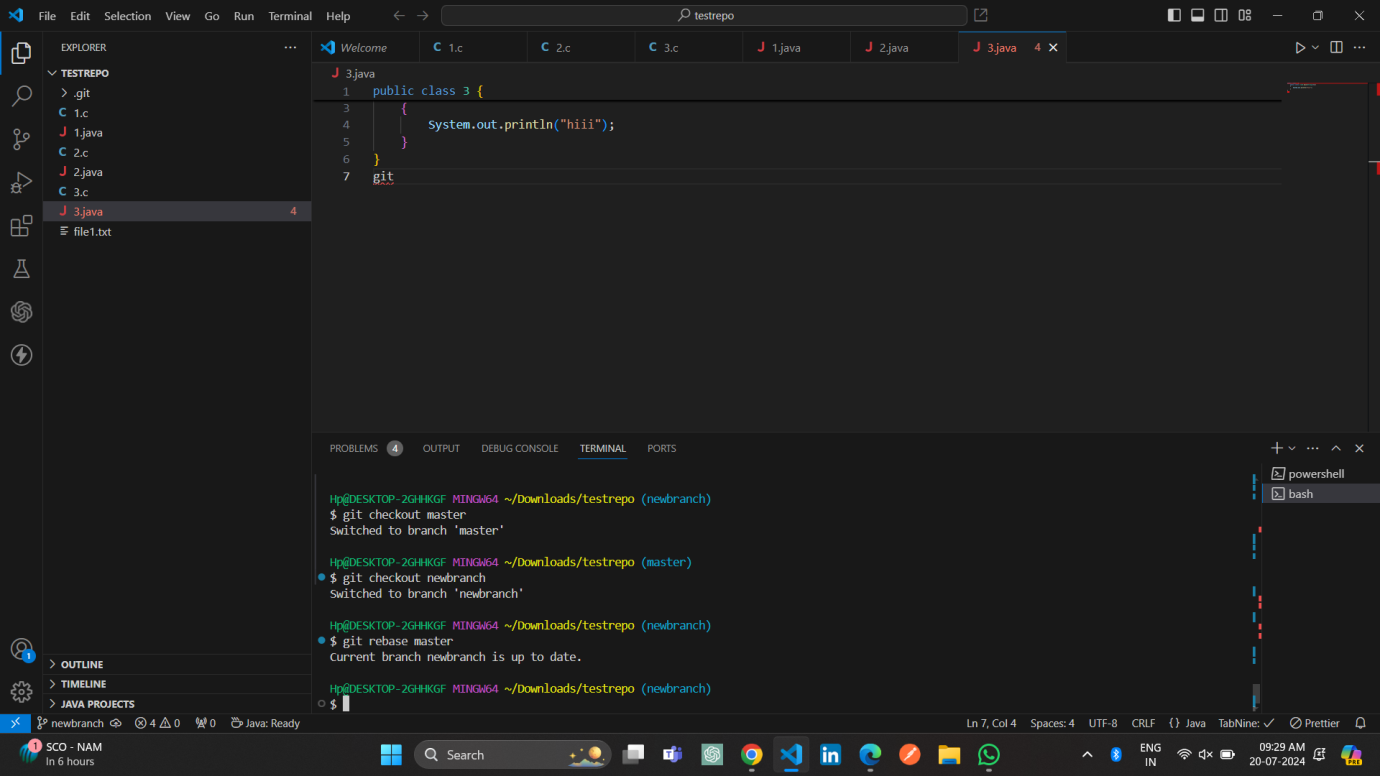
Create a branch. Make a commit.

**Step 6:**

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

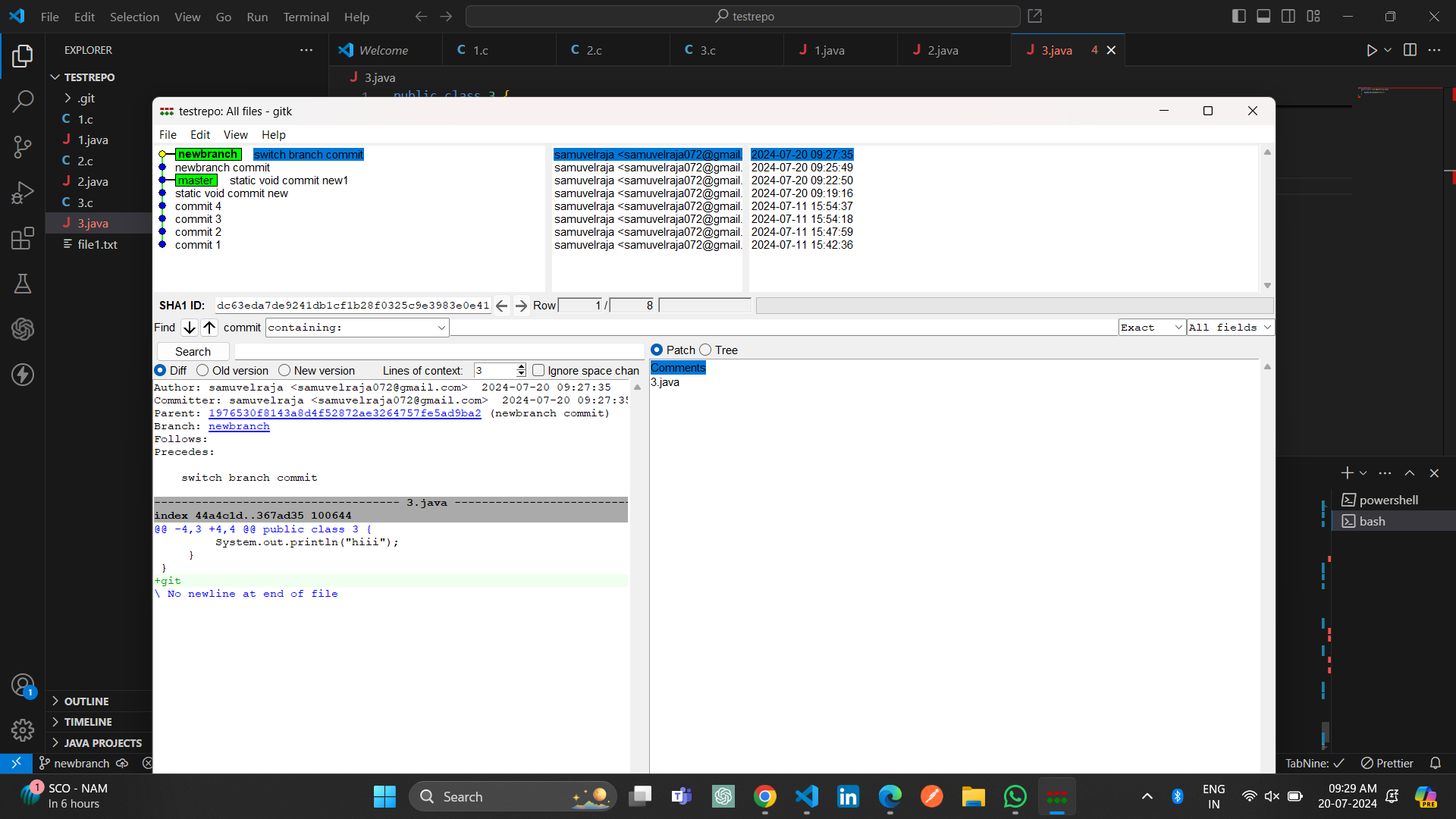
**Step 7:**

Now switch back to your branch. Use the rebase command in your branch.



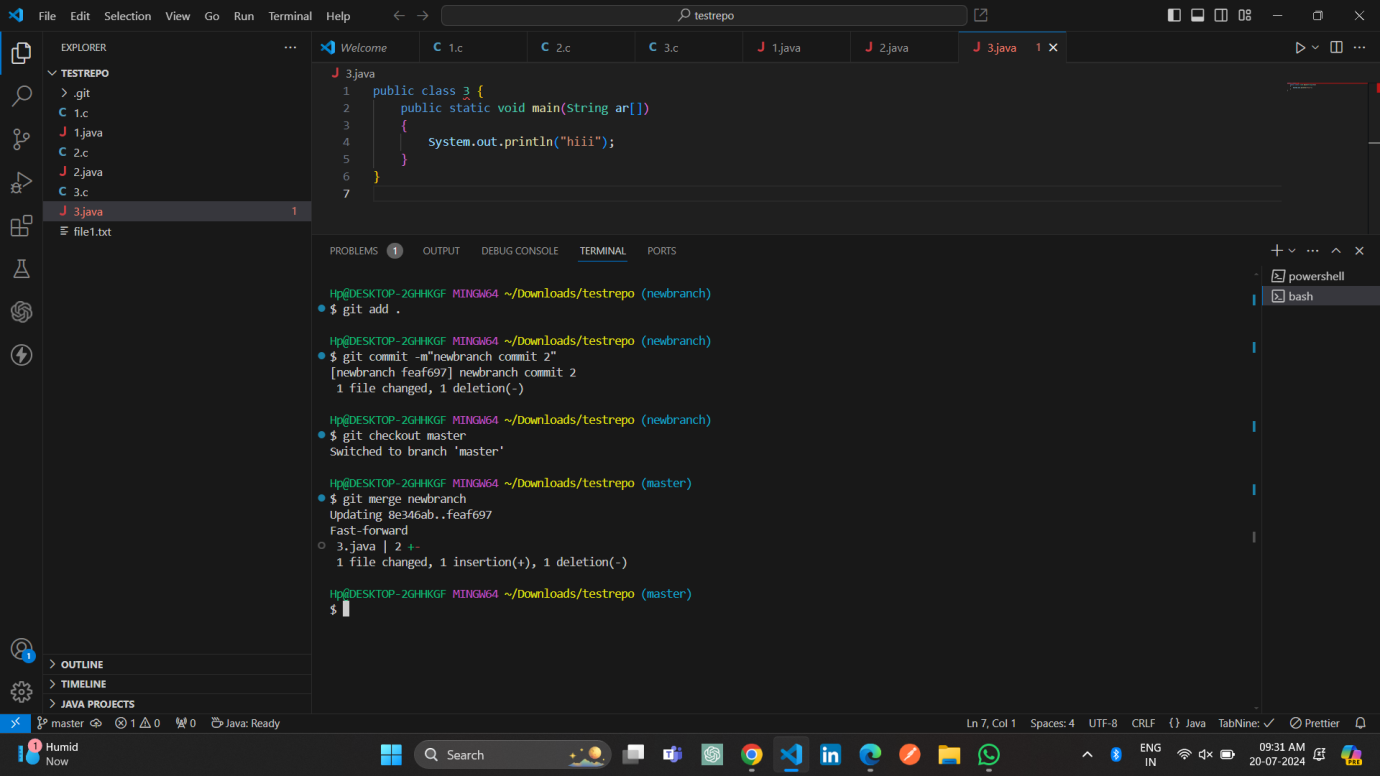
**Step 8:**

Look at the DAG in gitk, and note that you have the commit from the master branch, but no merge commit.

****

**Step 9:**

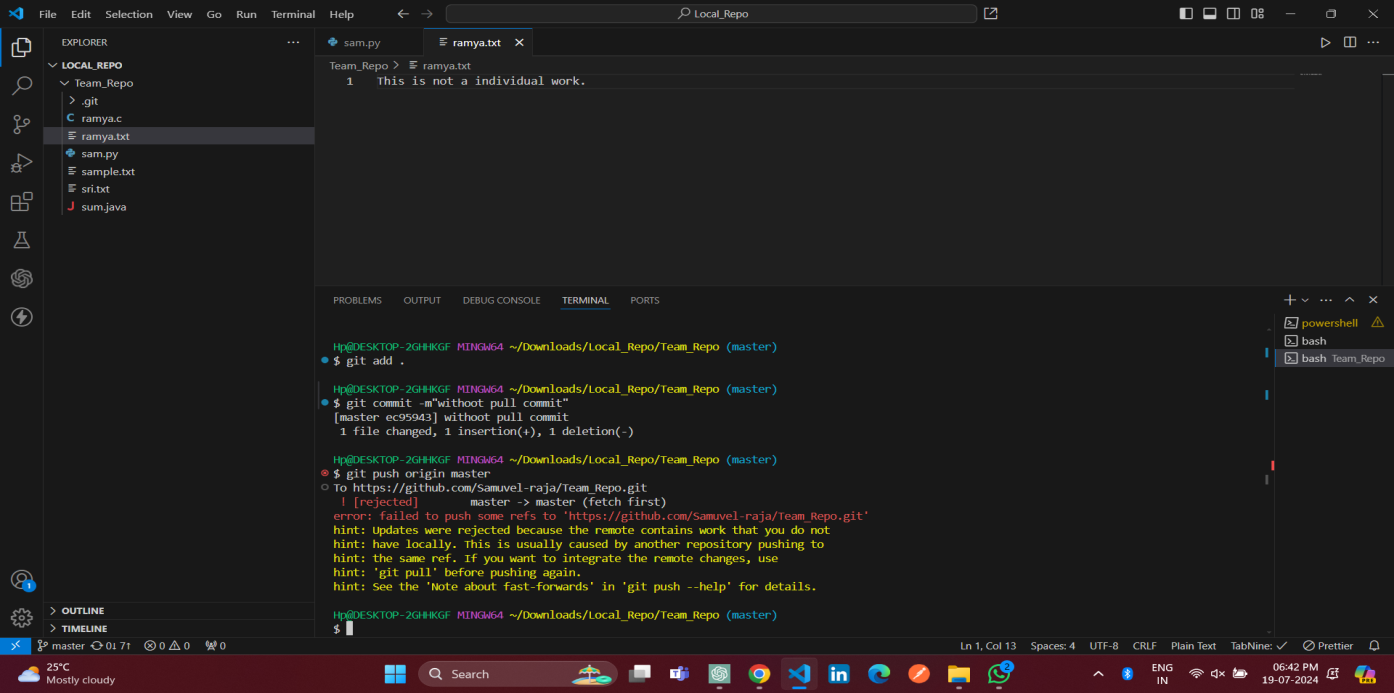
Make one more commit in your branch. Return to master. Merge your branch. Notice how, thanks to the rebase, this is a fastforward merge.

****

**Stretch task:**

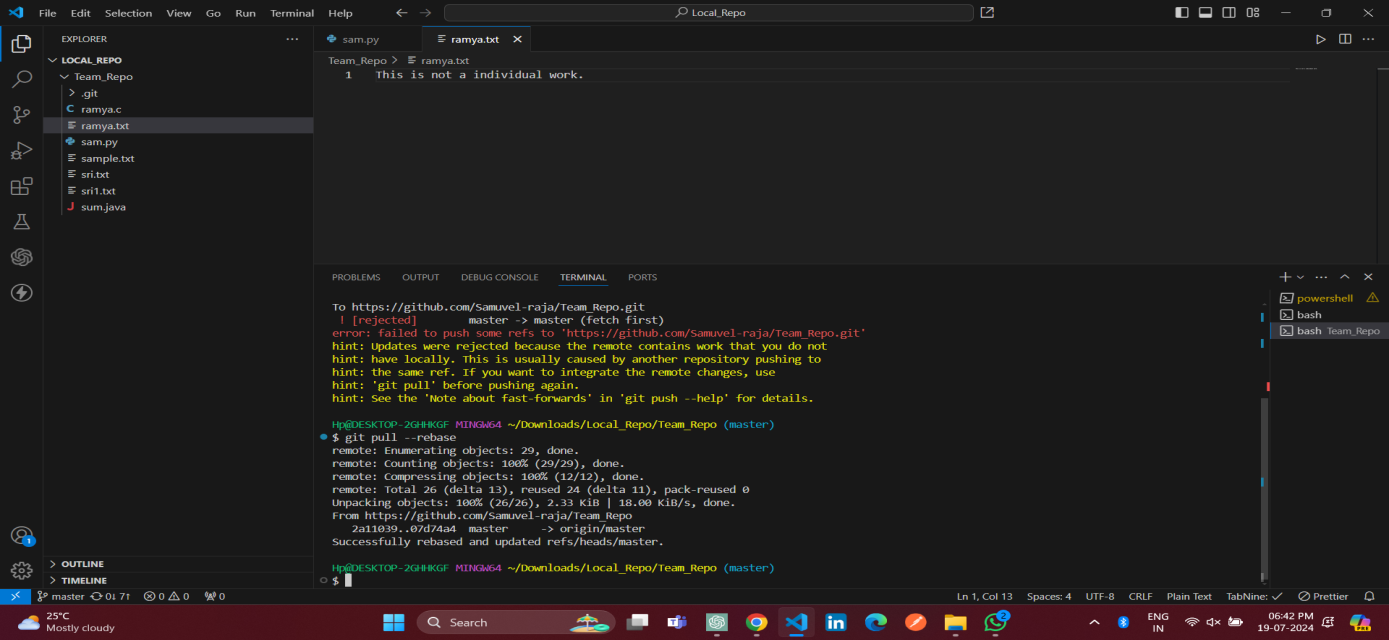
**Step 1:**

Find somebody from your team from the previous exercise. Have them push a commit to the central repository. Make a commit locally yourself also. Note that you should not have pulled their commit at this point. Try to push, and watch it fail.

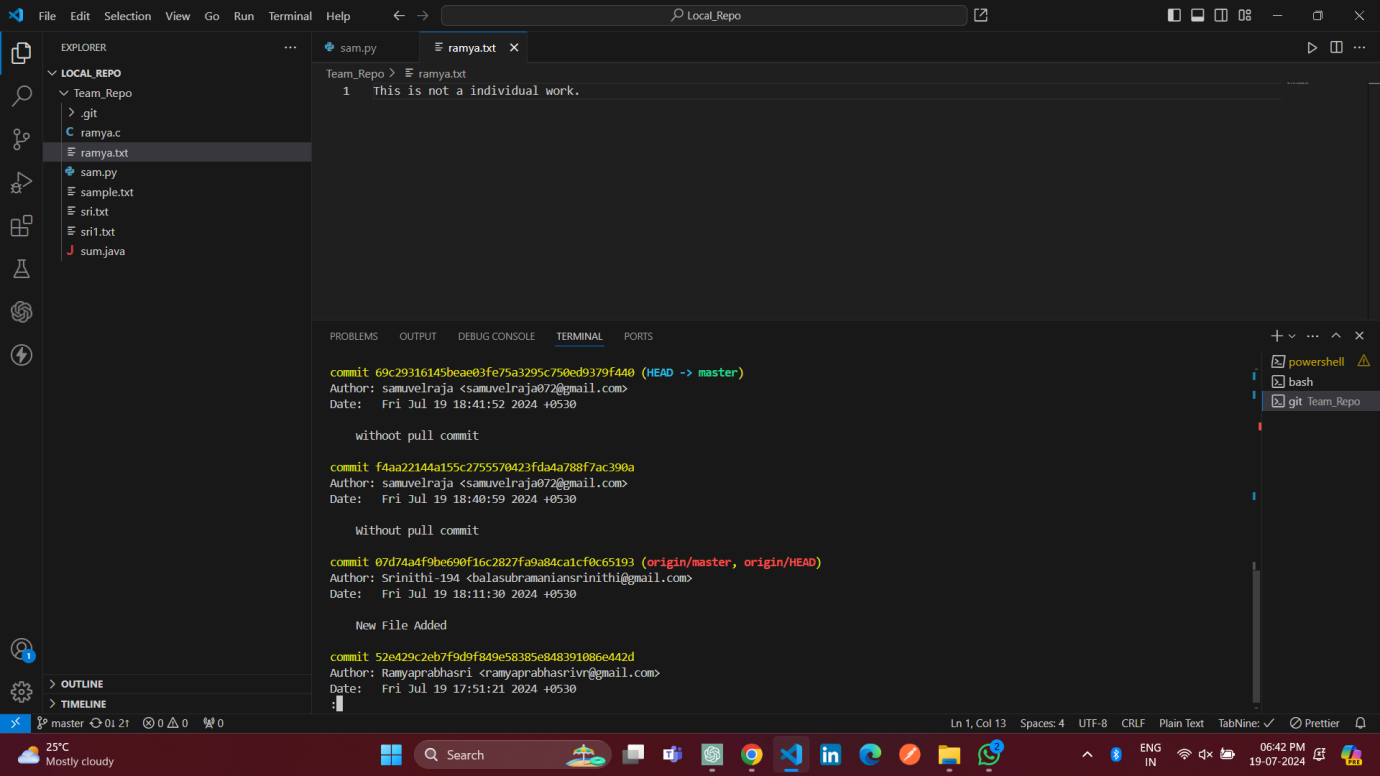
****

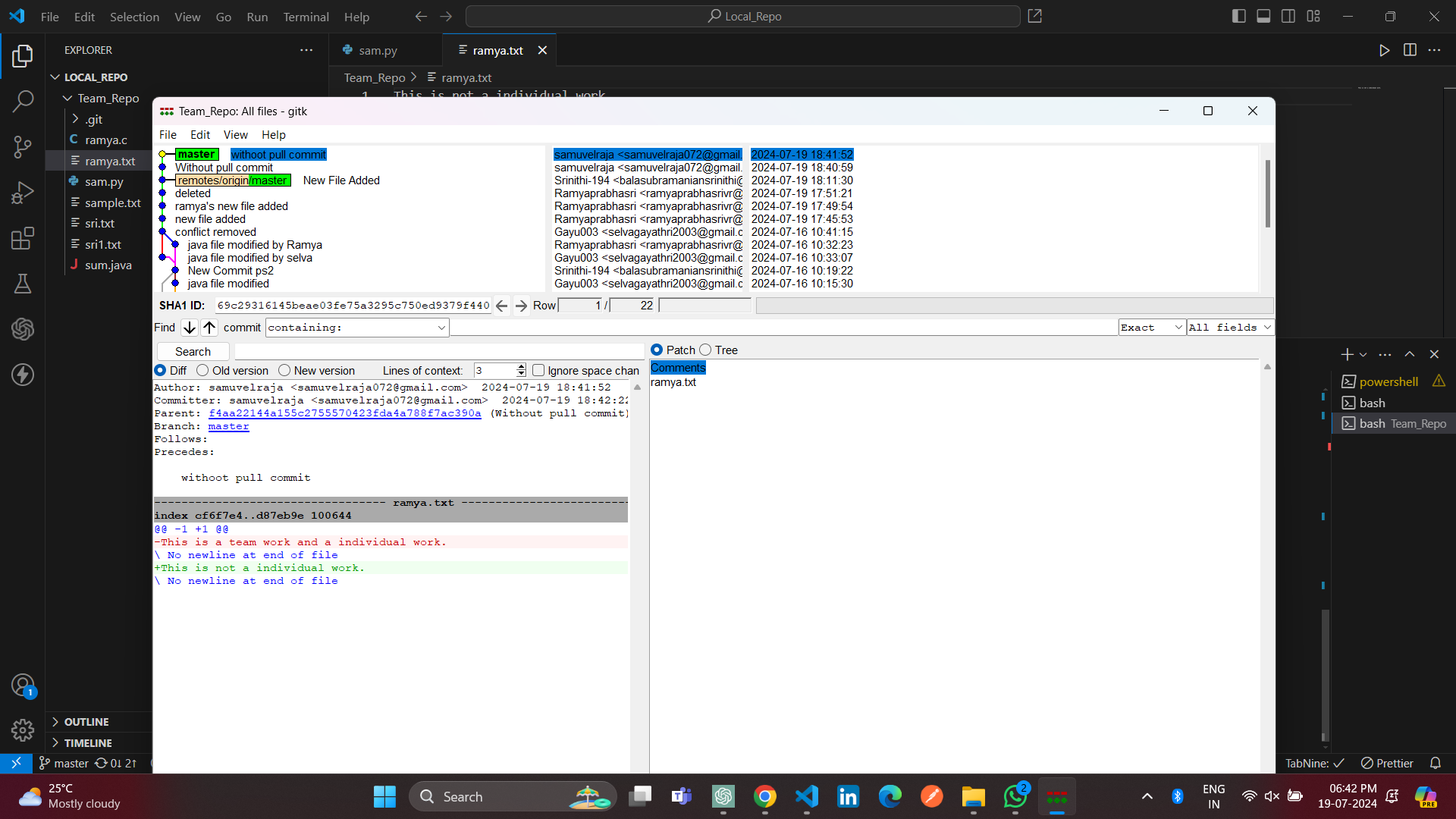
**Step 2:**

Now, pull but using the --rebase flag.

****

**Step 3:**

Use git log and gitk to verify that there is no merge commit, and the DAG is linear**.**Notice that your commit is the latest one, even though temporally the other member of your team made their commit afterwards.****

****