**1.QUESTION**

**GIT STASH:**

The git stash command **takes your uncommitted changes (both staged and unstaged), saves them away for later use, and then reverts them from your working copy**.

Here, we are doing

1.creating the file

2.modifying the file

3.git stash[takes the modified change and stores it]

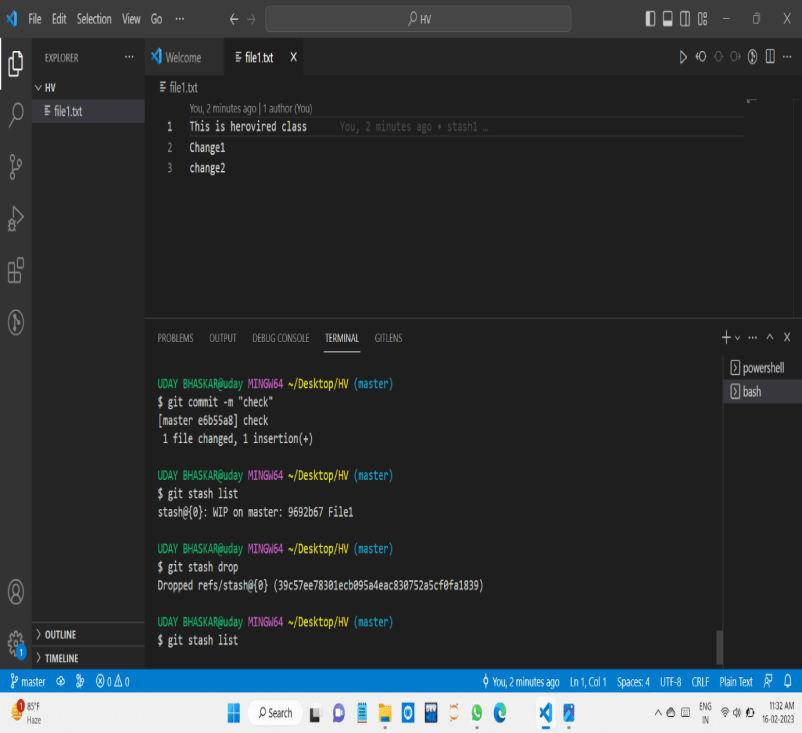
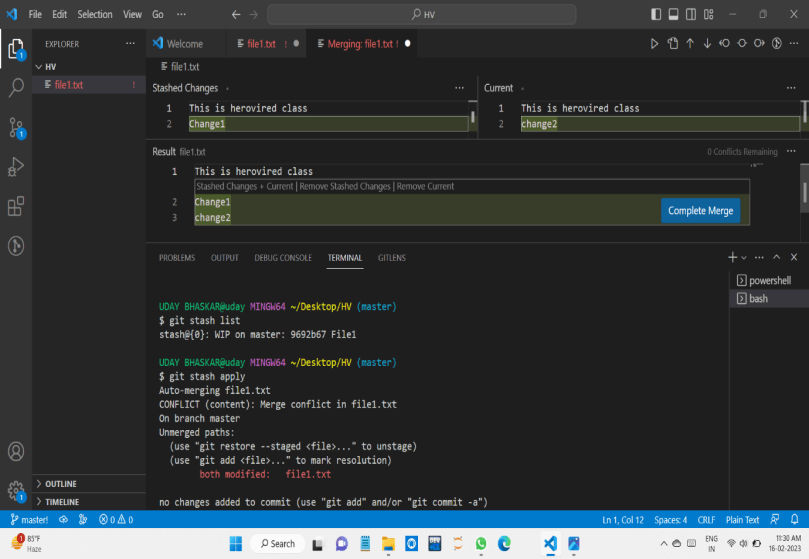
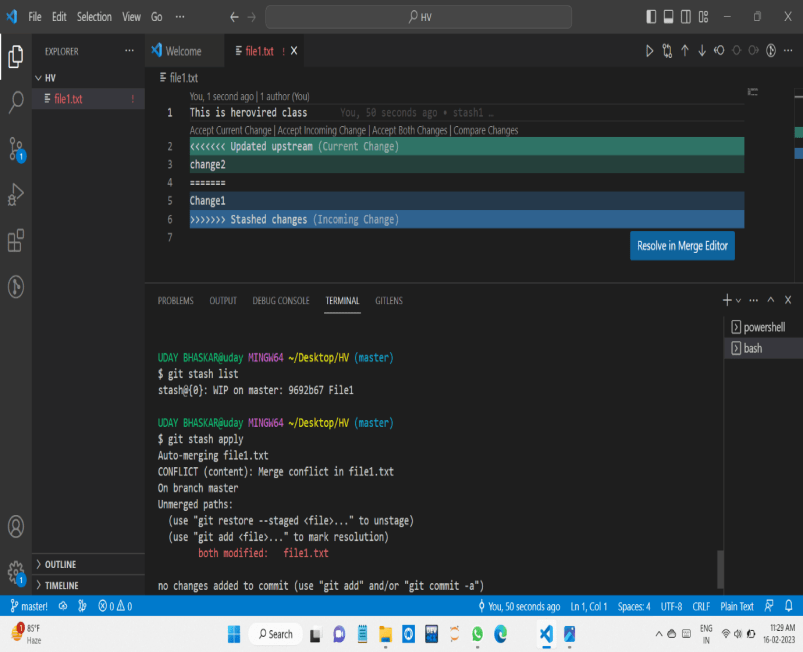
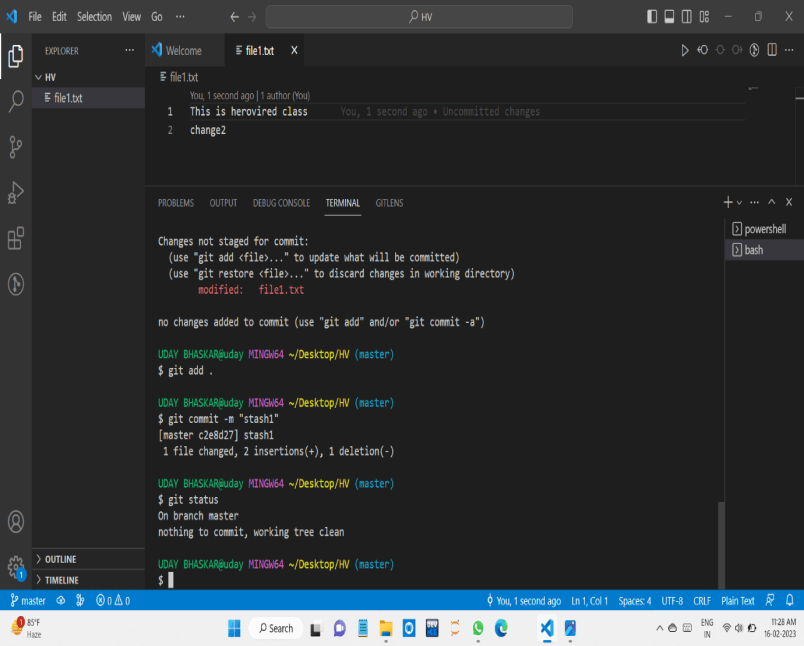
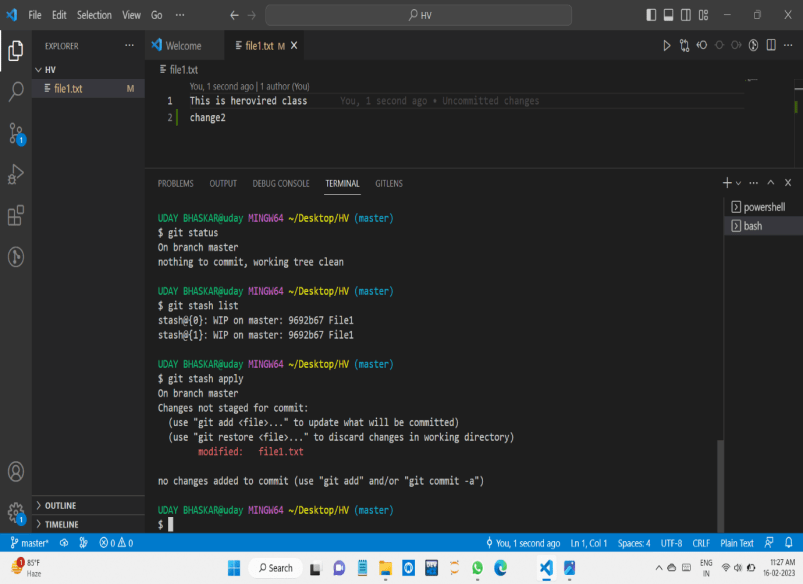
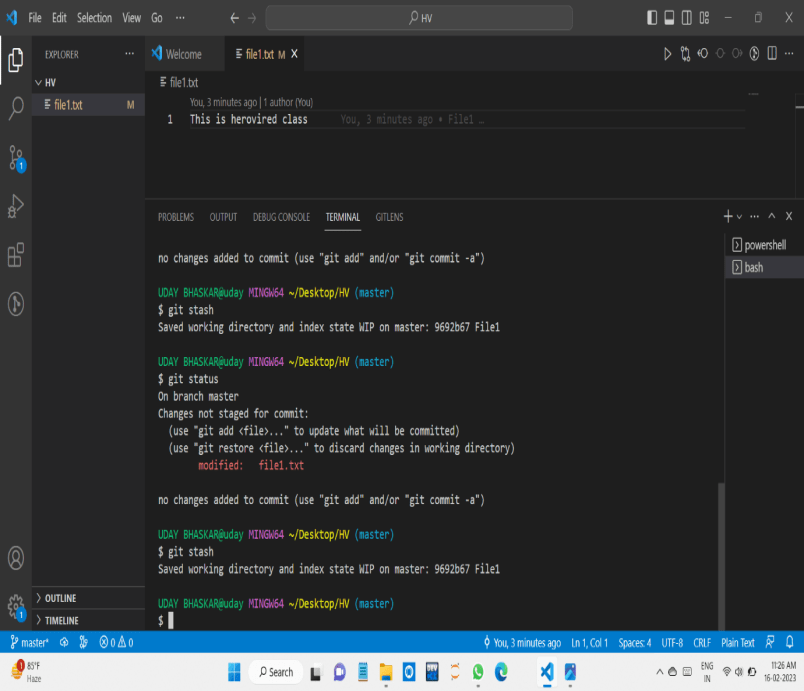
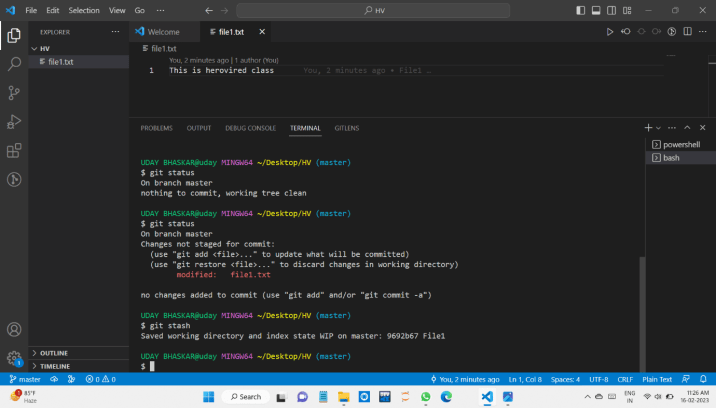
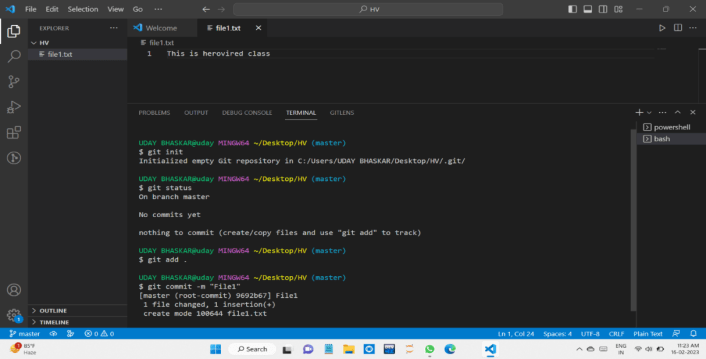
4.repeat 2,3[New change also stores it]

5.git stash list [shows all list in stash]

6.git stash apply so the top element in stash added to file then commit it.

7.similarli second one also we got an merge conflict and I solve it by combining together.

8.Also used git pop to remove top element in stash



**2.QUESTION**

**GIT FETCH:**

The "**git fetch**" **command** is used to pull the updates from remote-tracking branches. Additionally, we can get the updates that have been pushed to our remote branches to our local machines. As we know, a branch is a variation of our repositories main code, so the remote-tracking branches are branches that have been set up to pull and push from remote repository.

Here, we are doing

1.create a Remote for our local repository to remote repository[Herovired is our remote repository] and assign is our remote name here.

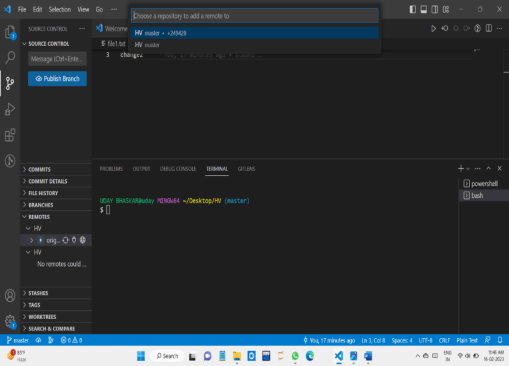
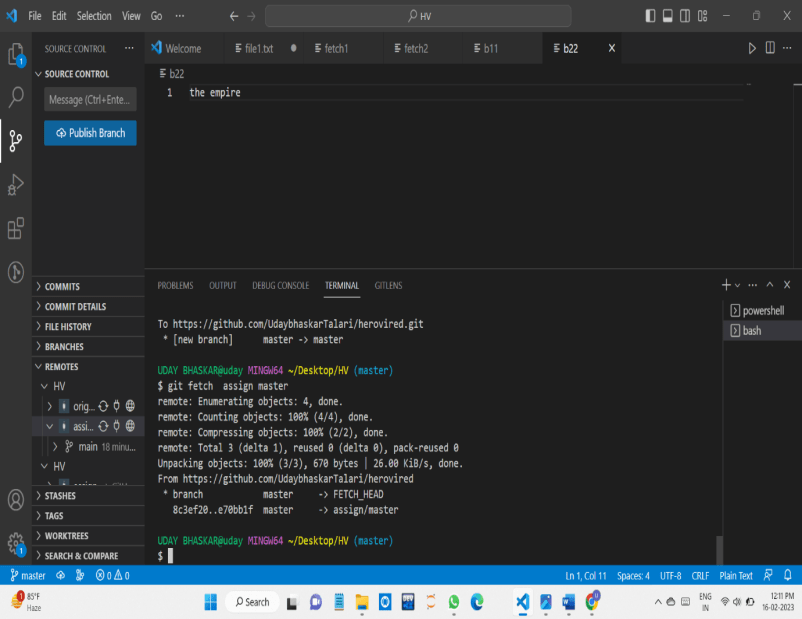
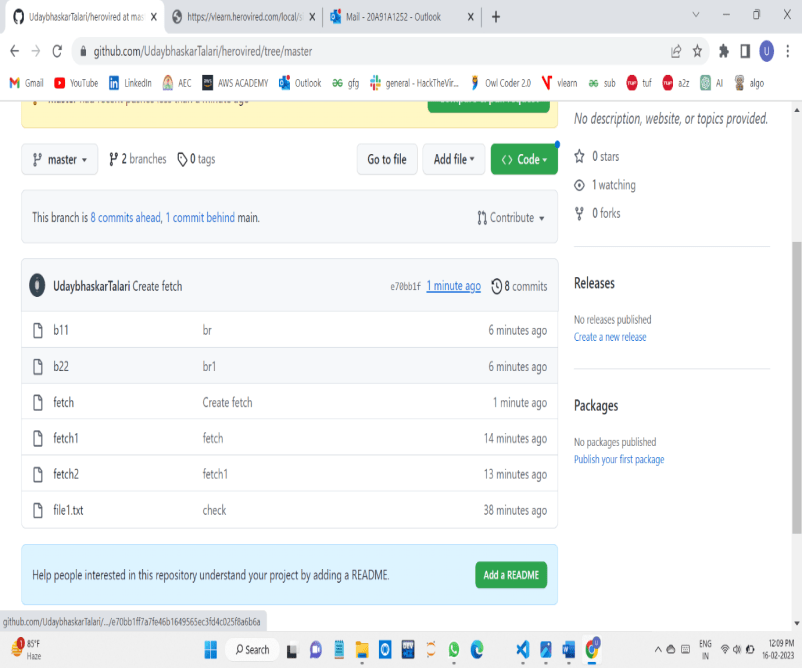
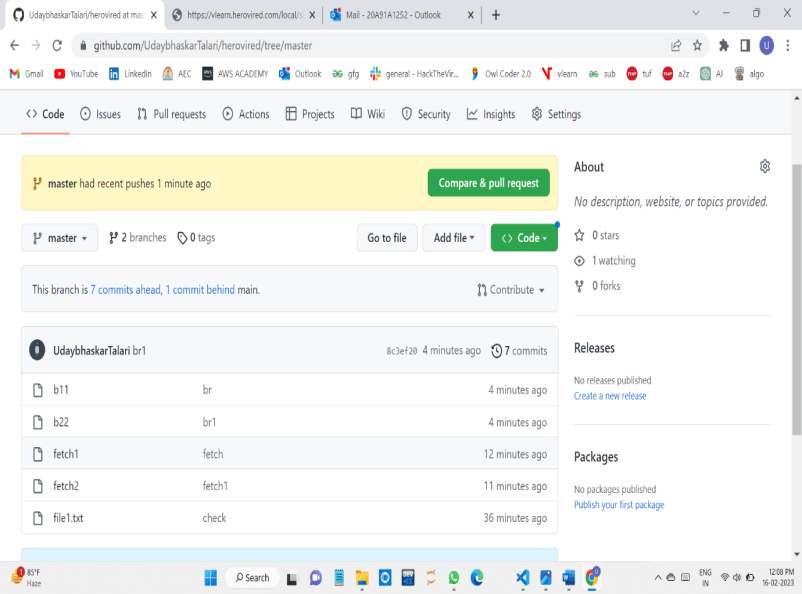
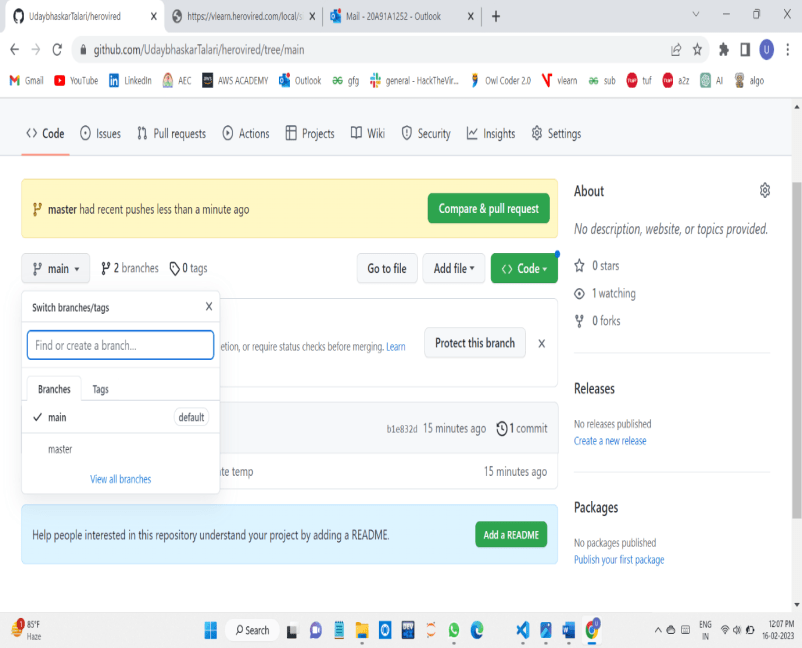
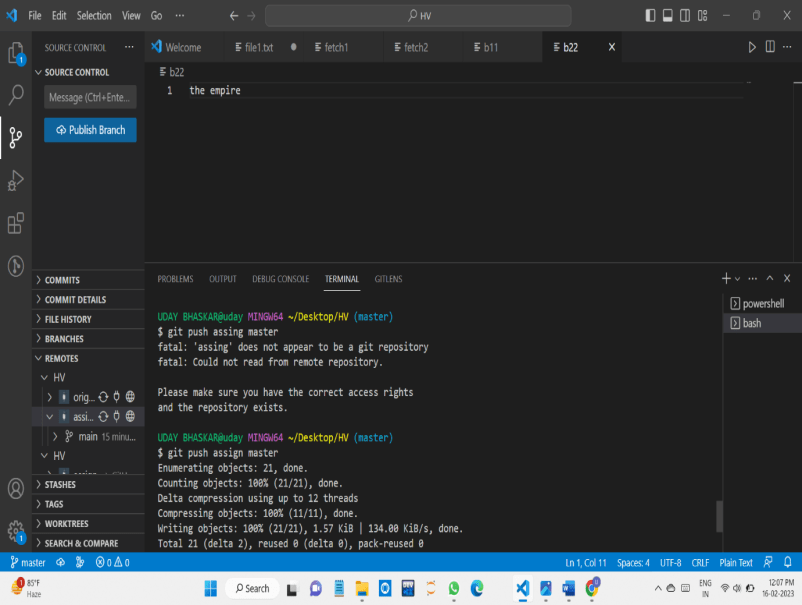
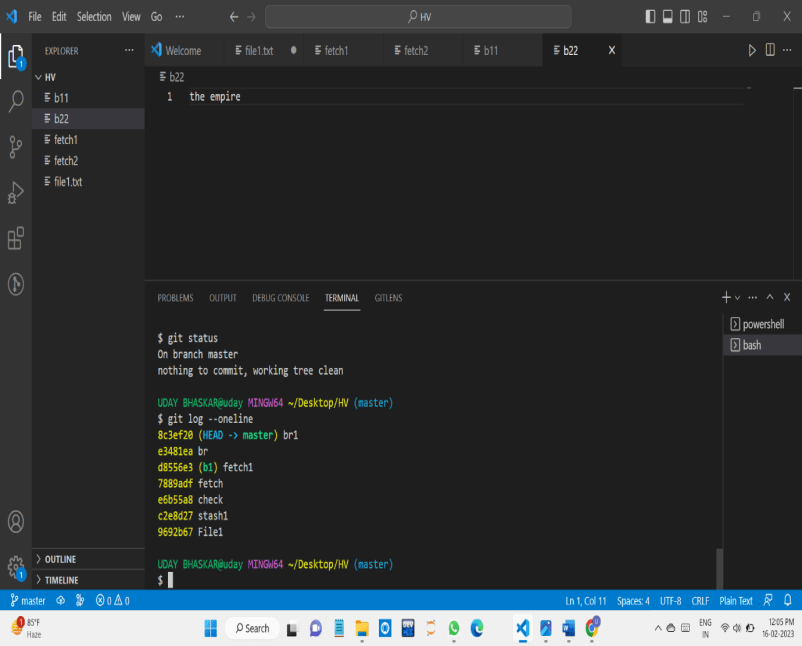
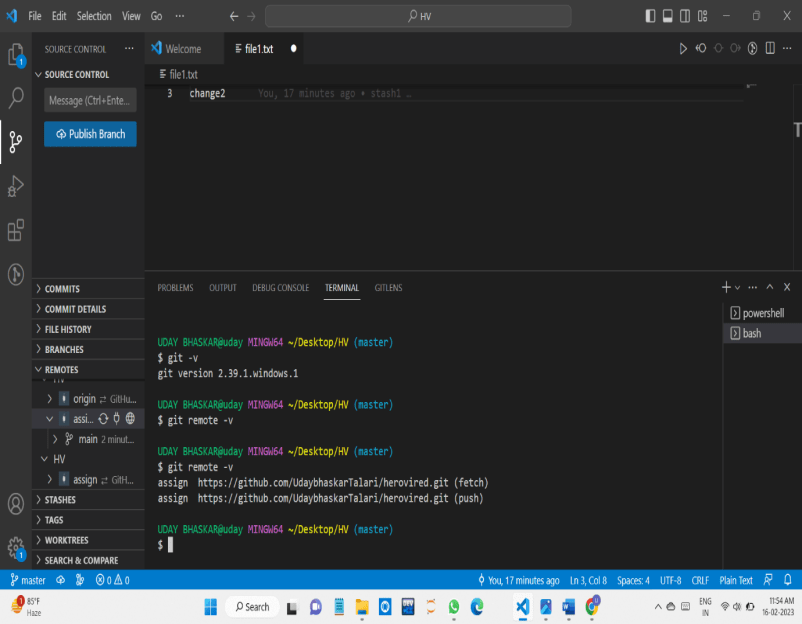
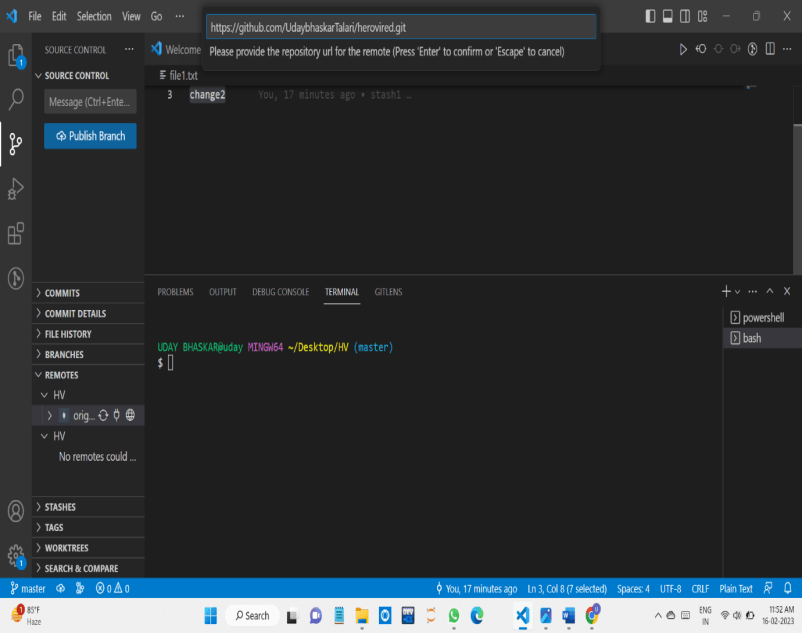
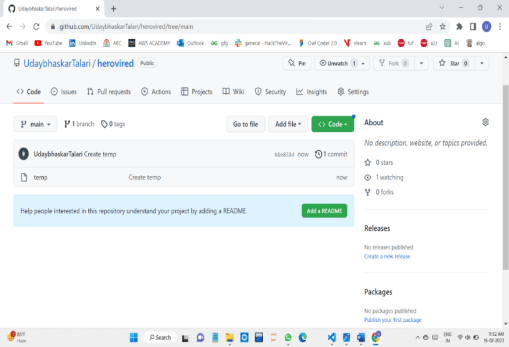
2.After that we created some sample files.

3.Push them to Herovired repository with the help of assign[our remote]

4.You can see them in the repository.

5.After that I made a change in one of the file.

6.I use git fetch remotename branchname [I got an an update what changes has been Made]

****

**GIT MERGE:**

In Git, the merging is a procedure to connect the forked history. It joins two or more development history together. The git merge command facilitates you to take the data created by git branch and integrate them into a single branch. Git merge will associate a series of commits into one unified history. Generally, git merge is used to combine two branches.

It is used to maintain distinct lines of development; at some stage, you want to merge the changes in one branch. It is essential to understand how merging works in Git.

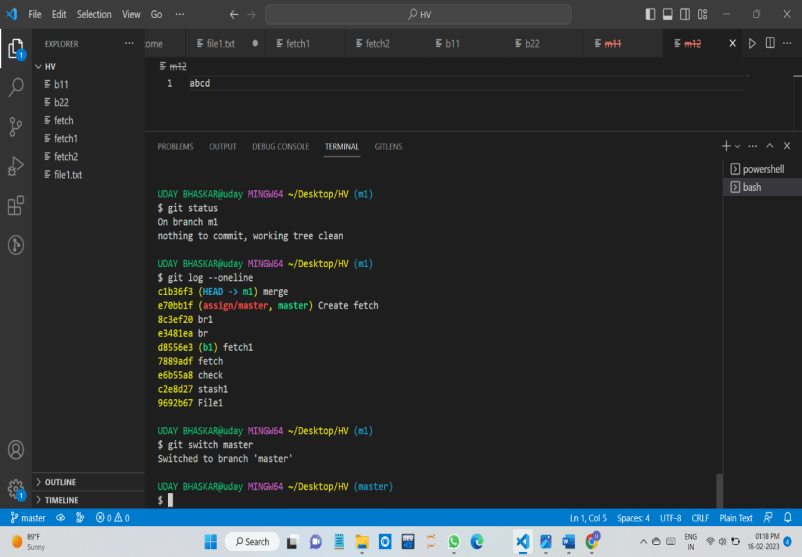
**STEPS:**

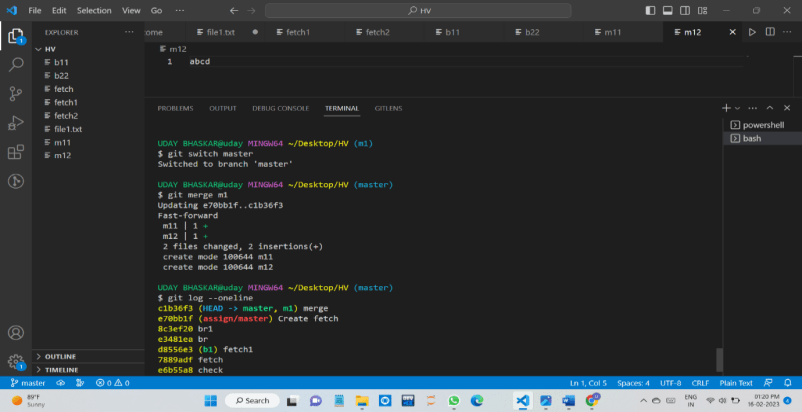
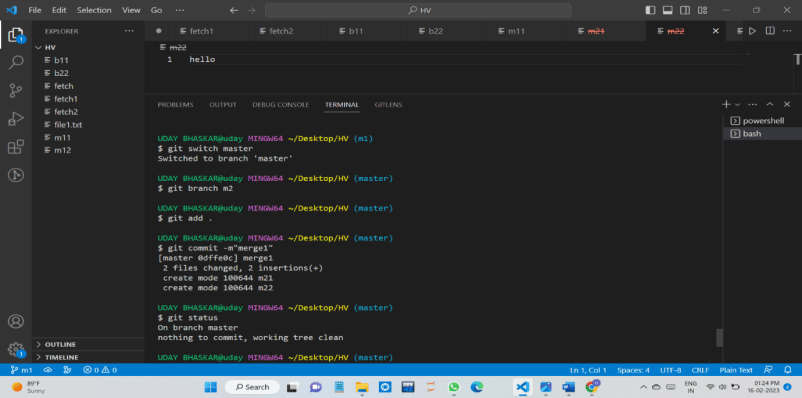
We are created a Two branches m1 and m2

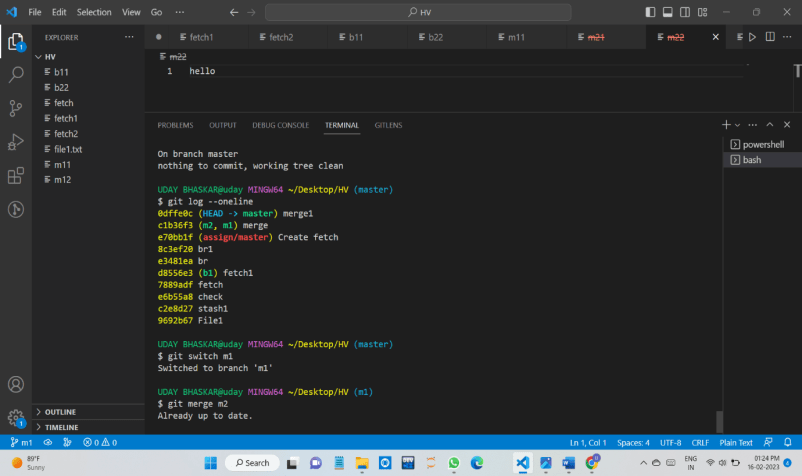
We create some files in both branches and we merge master and m1

We merge m2 with m1

They are at same level master level.







3.QUESTION

Difference Between GIT PULL AND FETCH

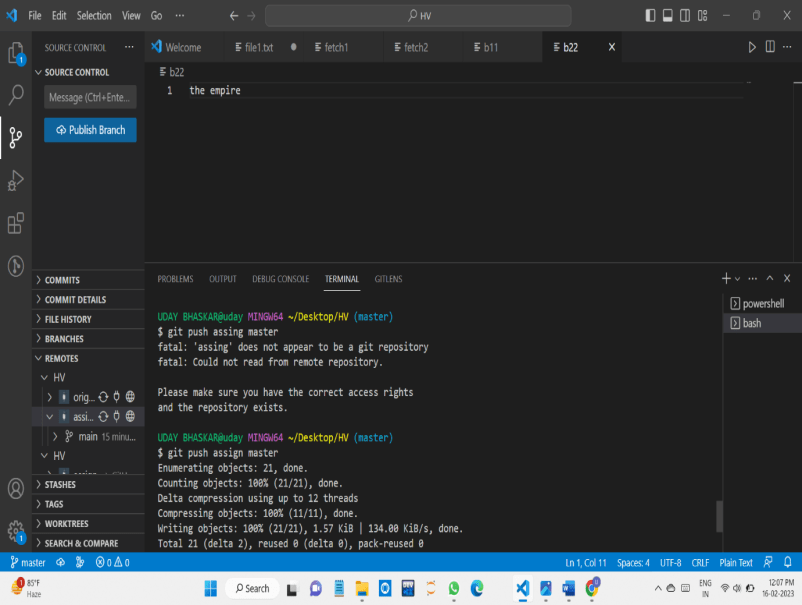
When comparing Git pull vs fetch, Git fetch is a safer alternative because it pulls in all the commits from your remote but doesn’t make any changes to your local files.

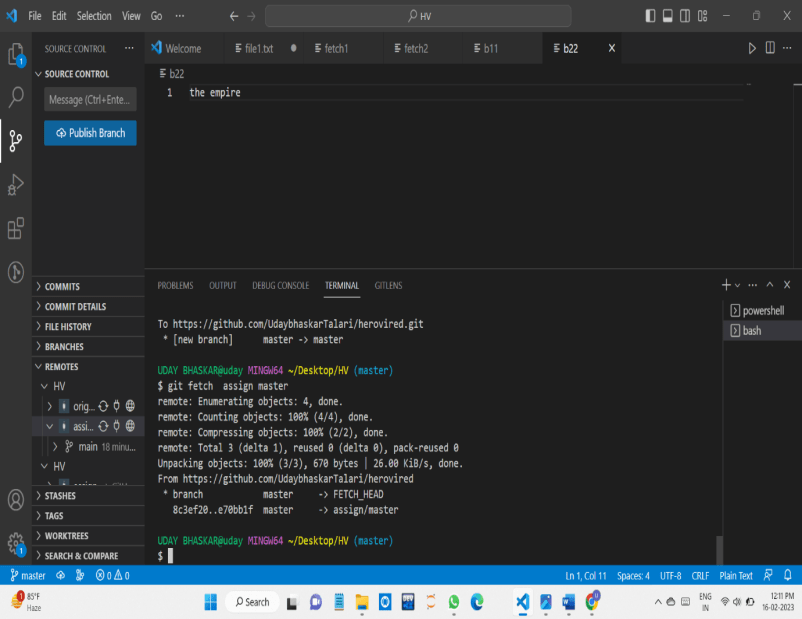
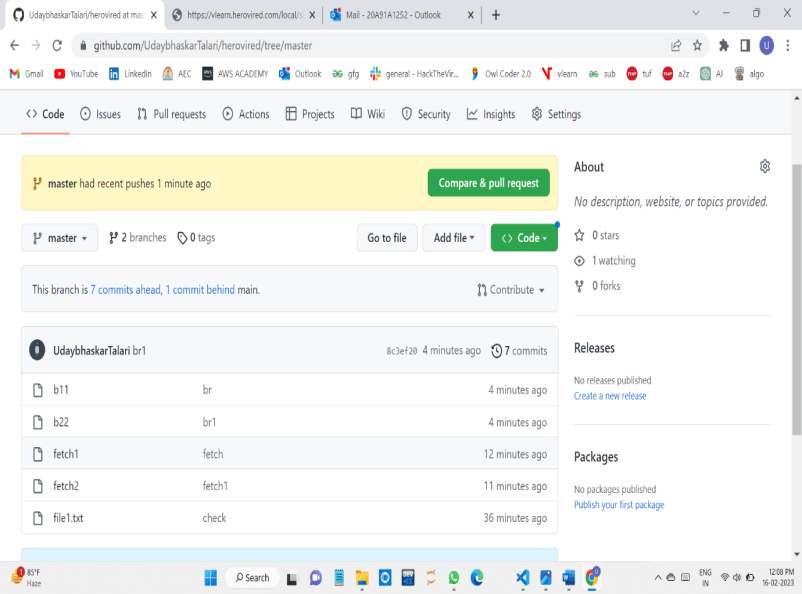
On the other hand, Git pull is faster as you’re performing multiple actions in one – a better bang for your buck. Using the Git pull command can be seen in one light as a feature of convenience; you’re probably less worried about introducing conflicts into your local repo and you just want the most up-to-date changes from the remote branch you’re pulling from.

Git pull is a more advanced action and it’s important to understand that you will be introducing changes and immediately applying them to your currently checked out branch.

GIT FETCH:

We push some files to the repository and modify them by using the fetch command we can pull them but don’t merge them.



 From the above we can see that we modify some files in remote repository and we can bring the to

Local but did not merge them.

**GIT PULL:**

**The git pull command is used to fetch and download content from a remote repository and immediately update the local repository to match that content.**

**The git pull command is actually a combination of two other commands, git fetch followed by git merge**

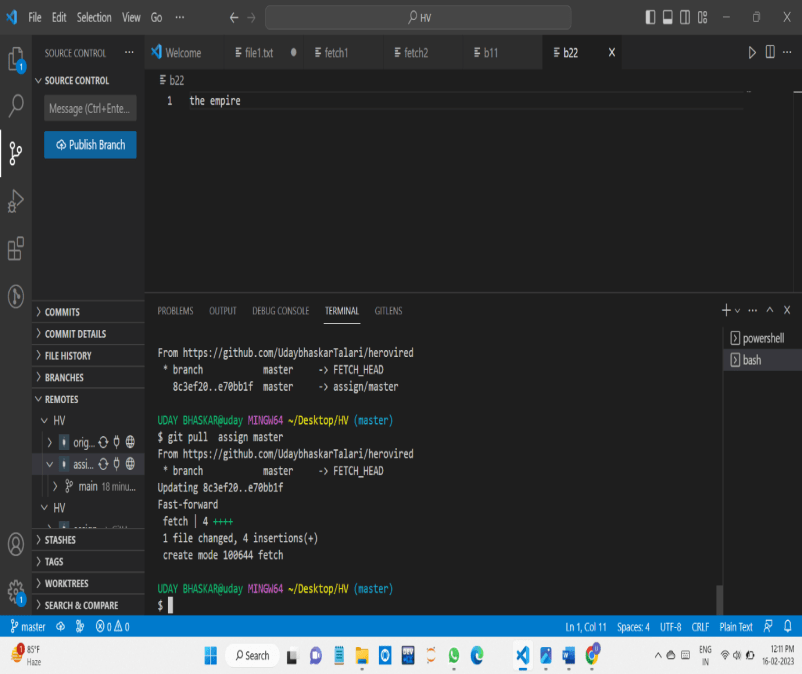
**The git pull command first runs git fetch which downloads content from the specified remote repository. Then a git merge is executed to merge the remote content refs and heads into a new local merge commit.**

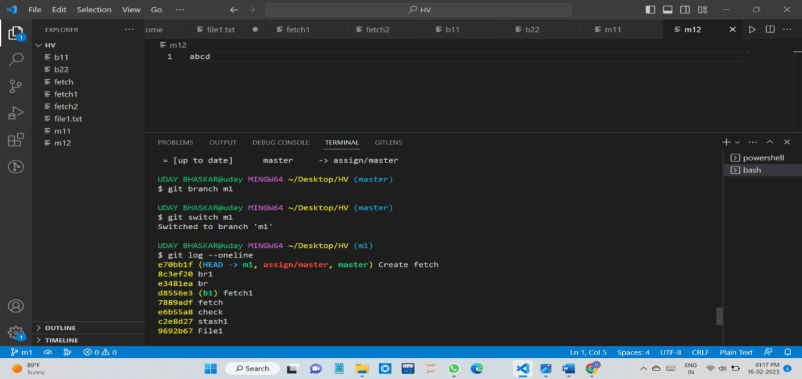
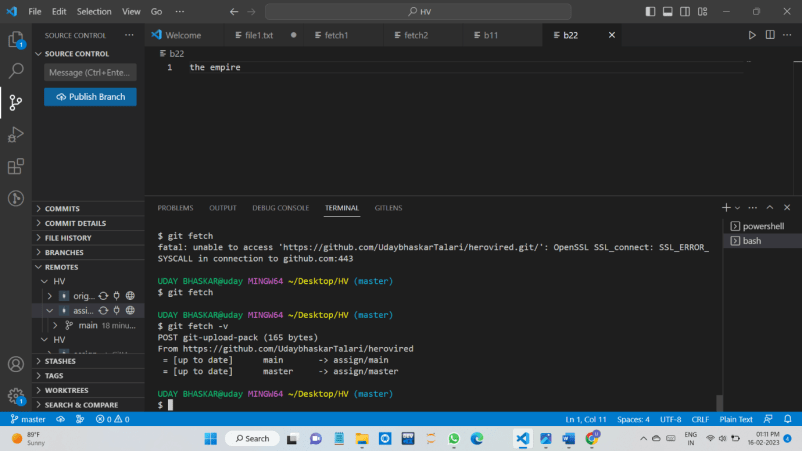
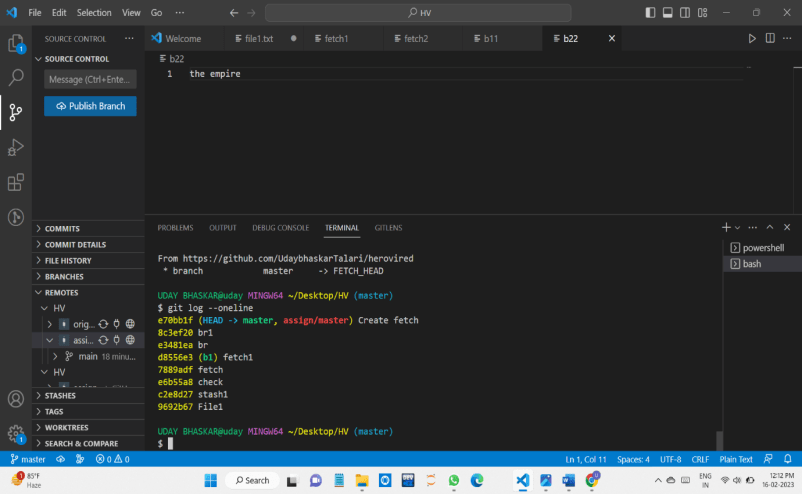
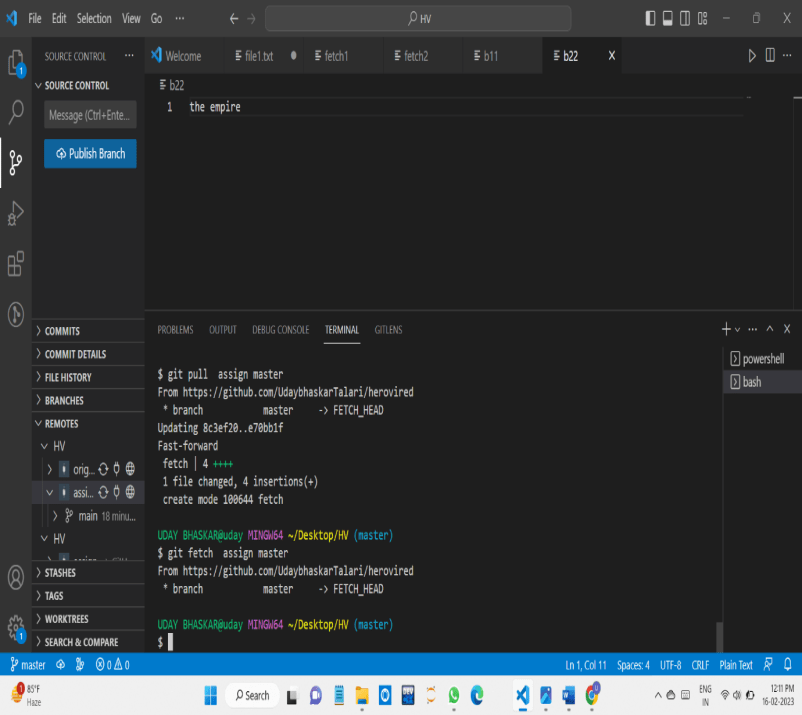
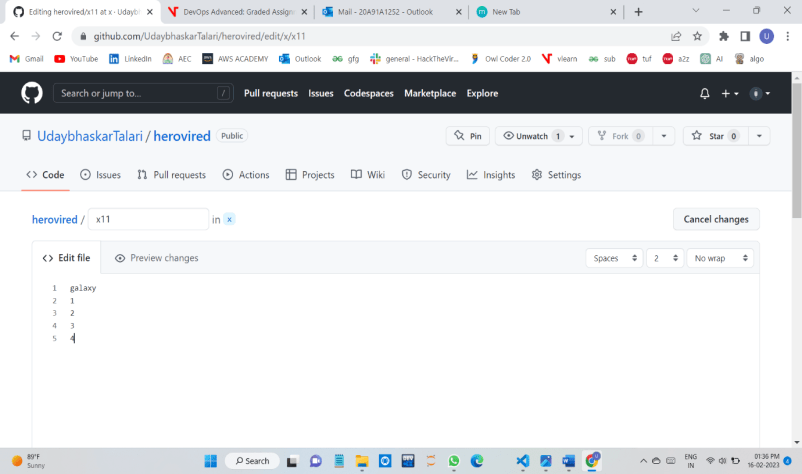
**After Fetch we came to know that there is an update in files in repository**

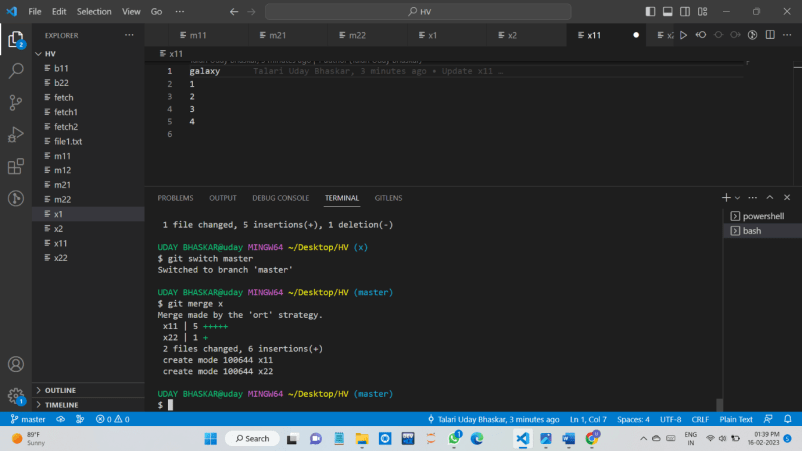
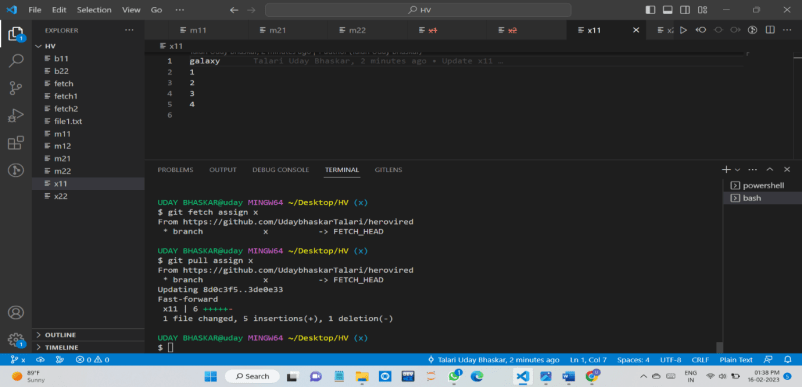
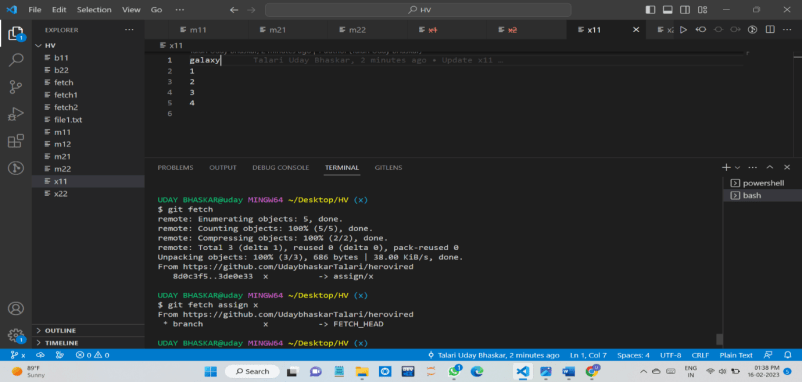
**So we Used the git pull to fetch the updates.**

**As well as merged to the local file we got the modified file**

**Here it fetches the updates and merge the files also.**

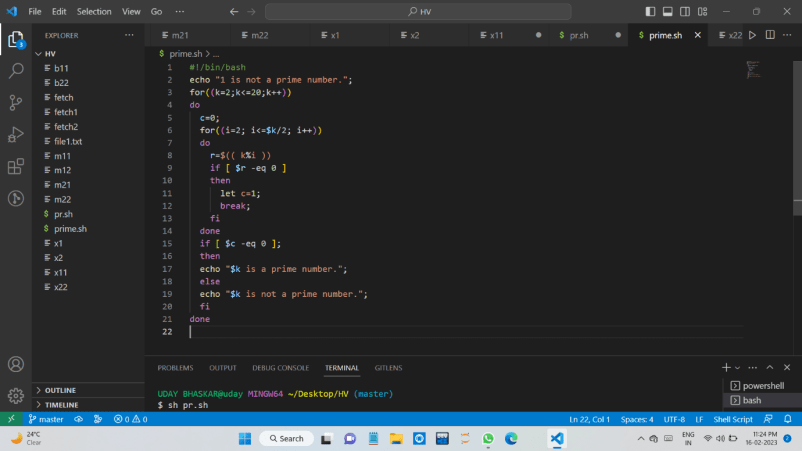


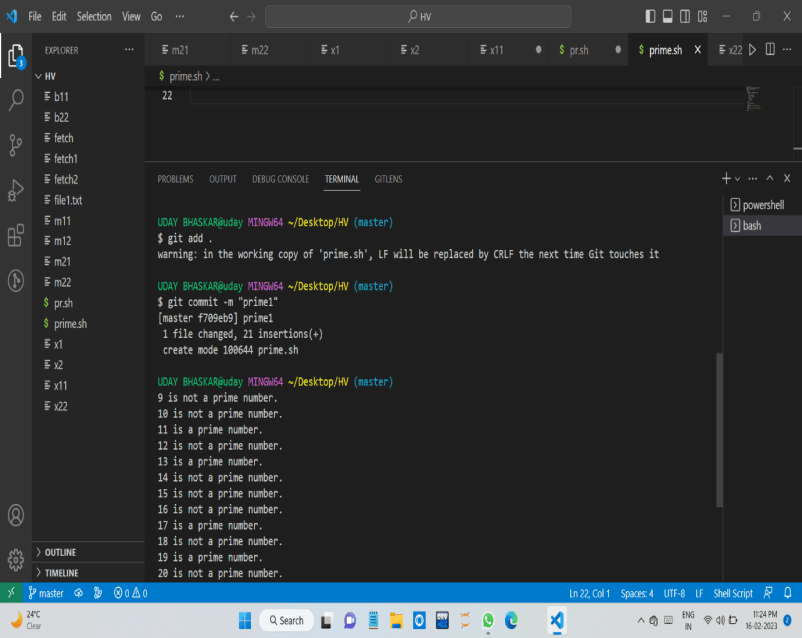
After pull command we can see that if we use the fetch there is nothing to update. 

If the modified file is above one in the remote repository this can be seen in local repository by using pull command

**4.QUESTION**

**BASH SCRIPT FOR 1 TO 20 PRIME NUMBERS.**





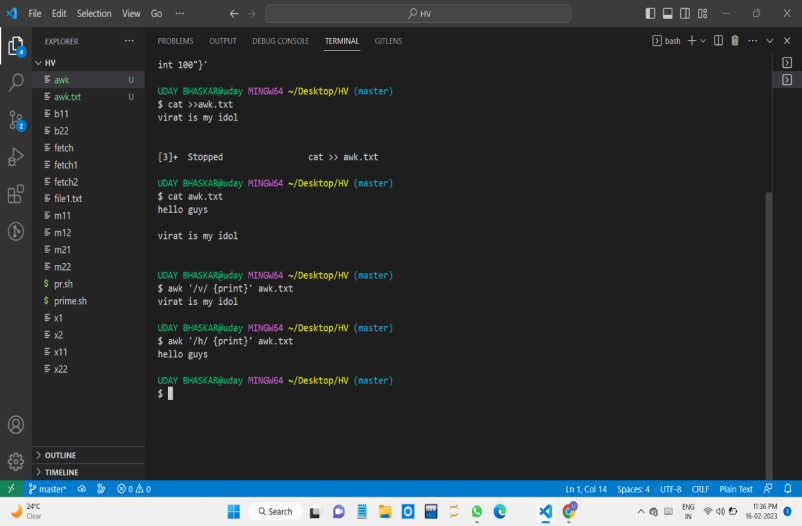
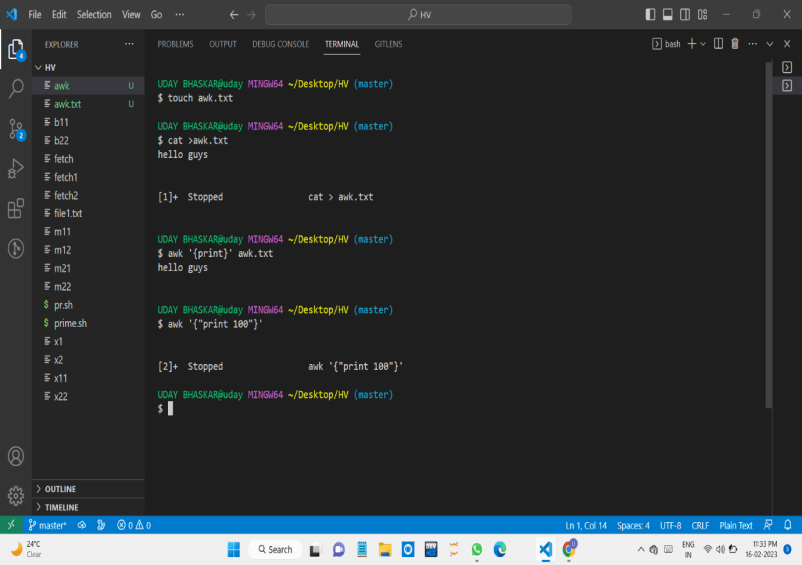
**AWK:**

\*The awk command is used for text processing in Linux.

\*Syntax:

awk options 'selection \_criteria {action }' input-file > output-file

\*awk command is used to filter the data in files.



**5.QUESTION**

**Set up a container and run a Ubuntu operating system**

**1.IMAGE**

**\*It is a read only binary template to create a containers**

**\*if is like a seed that contains the basic snapshots**

**it contains the 1.file system snapshots**

**2.starting command**

**2.CONTAINER**

**\*containerisation is a process,docker is based on this**

**\*it provides a virtual neutral space to run and execute command**

**\*it is used to hold the entire package**

**After downloading of that we can see in docker.**

