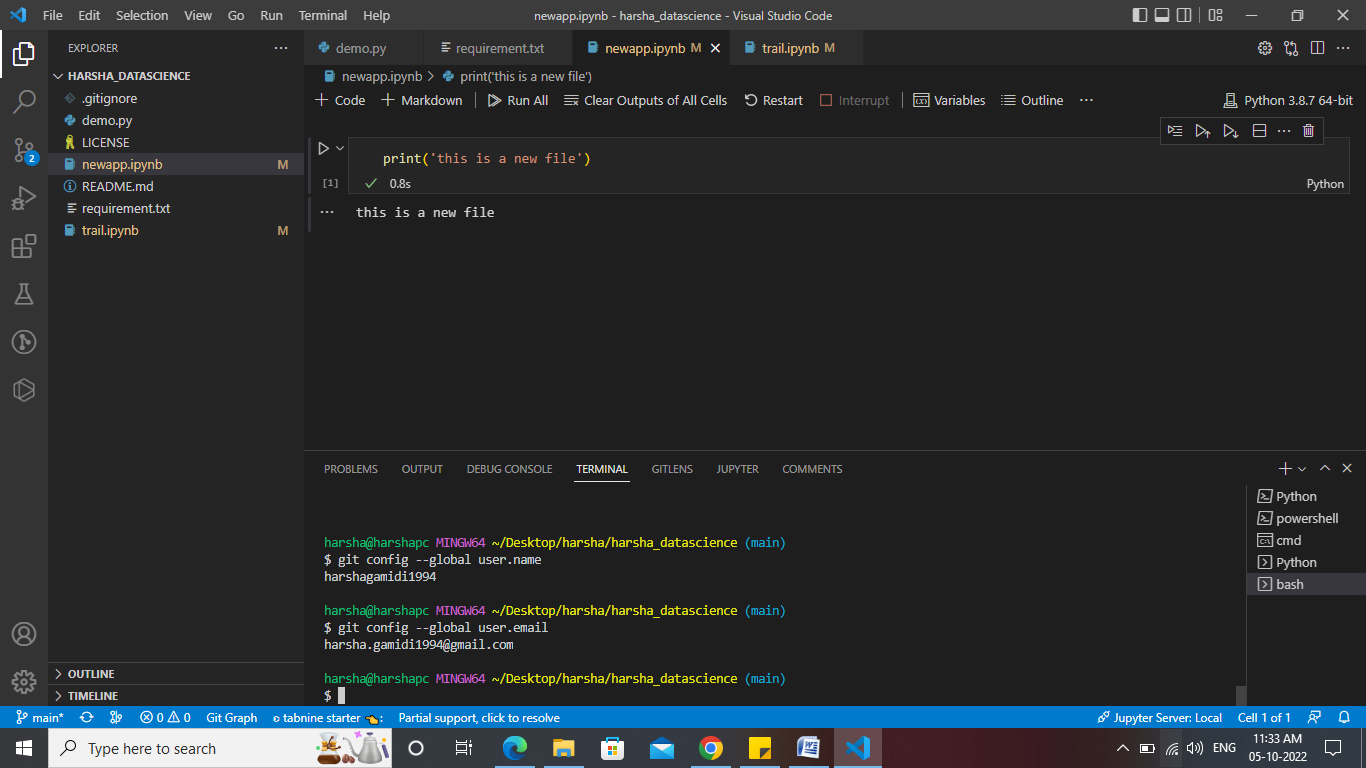
**GIT COMMANDS AND EXPLANATION**

**1. git config**

This command sets the author name email and password which will be used for all the commits made by the a particular user in the local repository

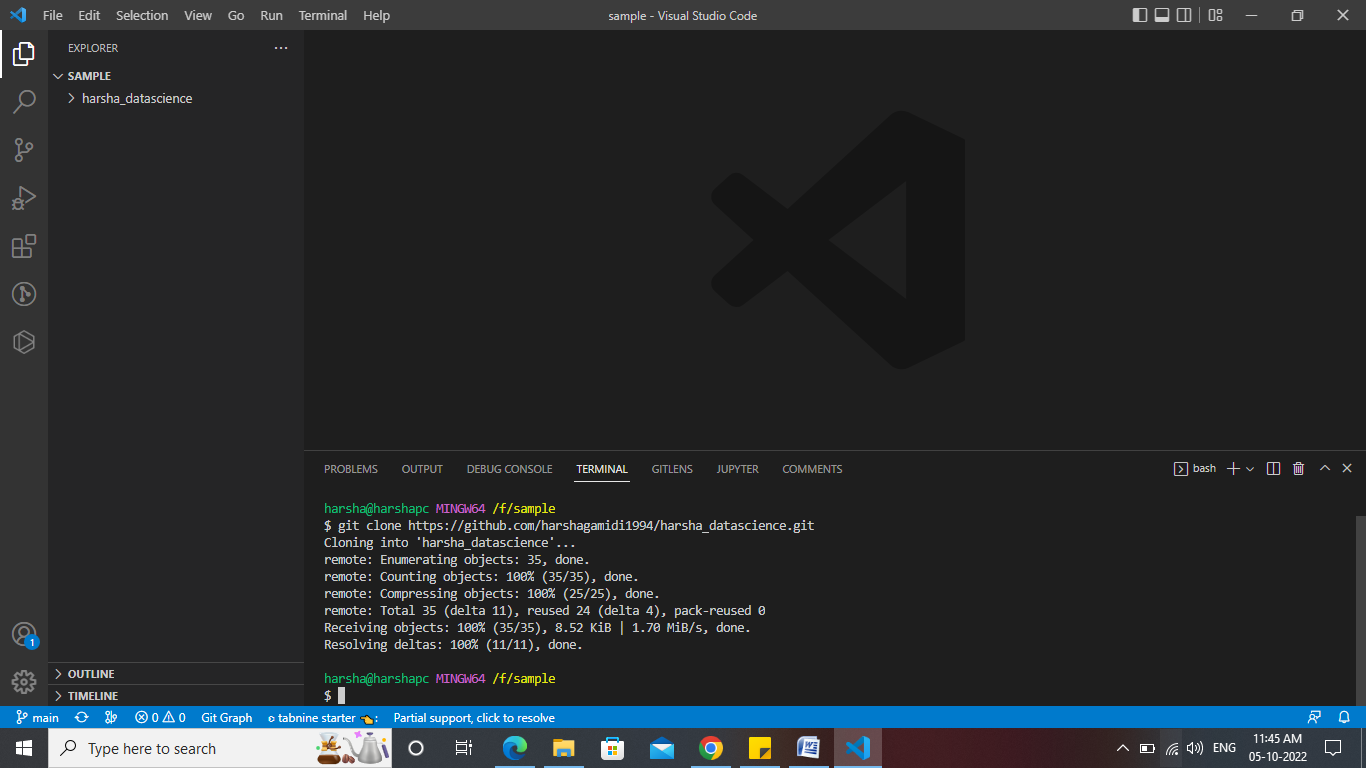
**Screenshot:**

****

**2.** **git clone**

This command helps us to make a copy of the repository in local system from the mentioned URL in the Github account

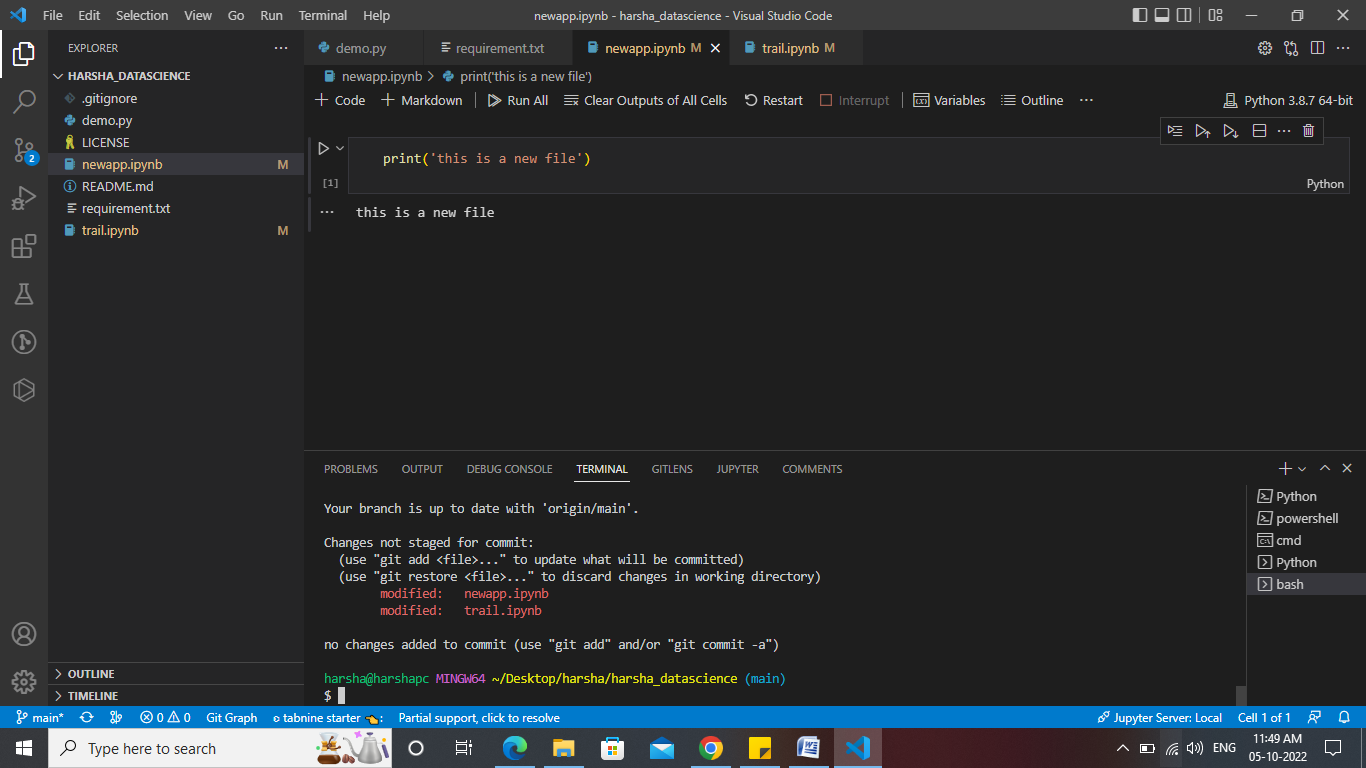
**Screenshot:**



**3. git status**

The status command is used to display the state of the working directory and the staging area. It allows you to see which changes have been staged, which haven't, and which files aren’t being tracked by Git.

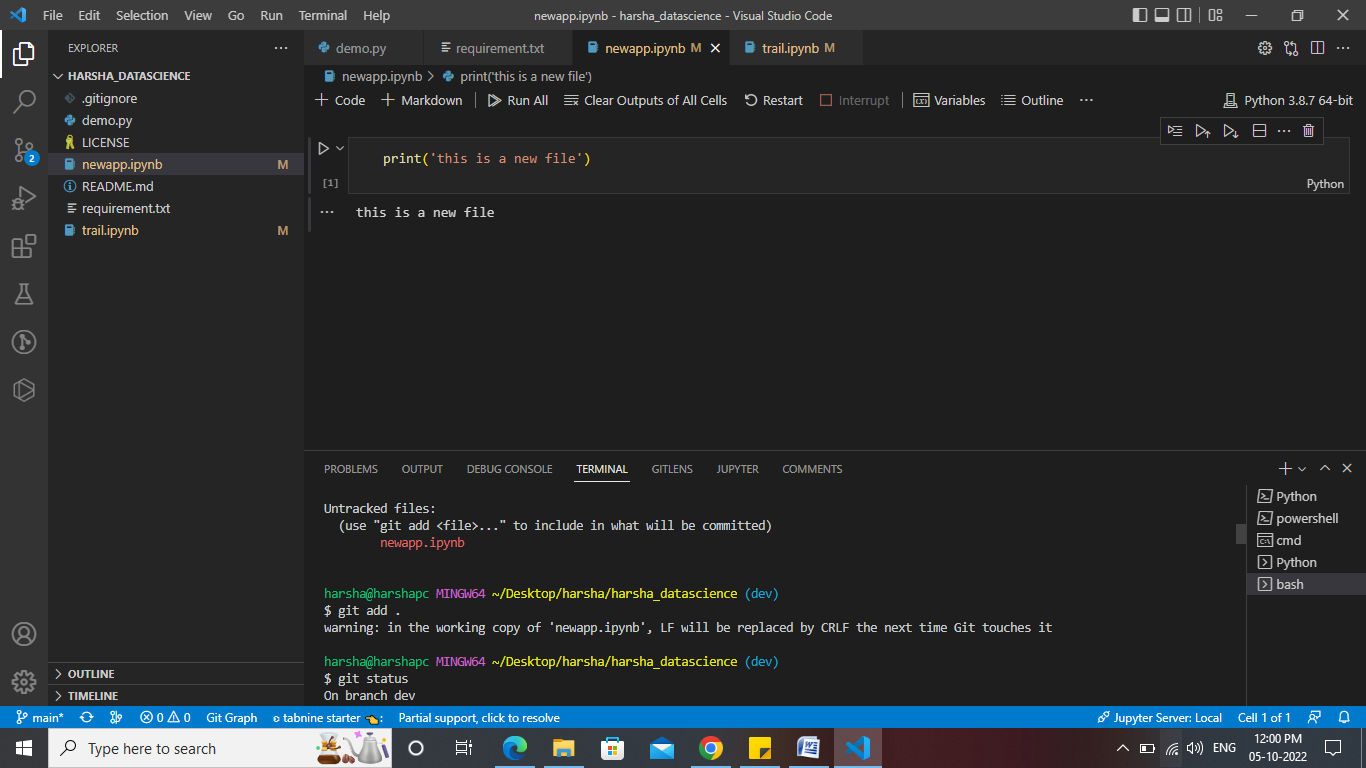
**Screenshot:**



**4. git add**

This command is used to add one or more files to staging (Index) area.

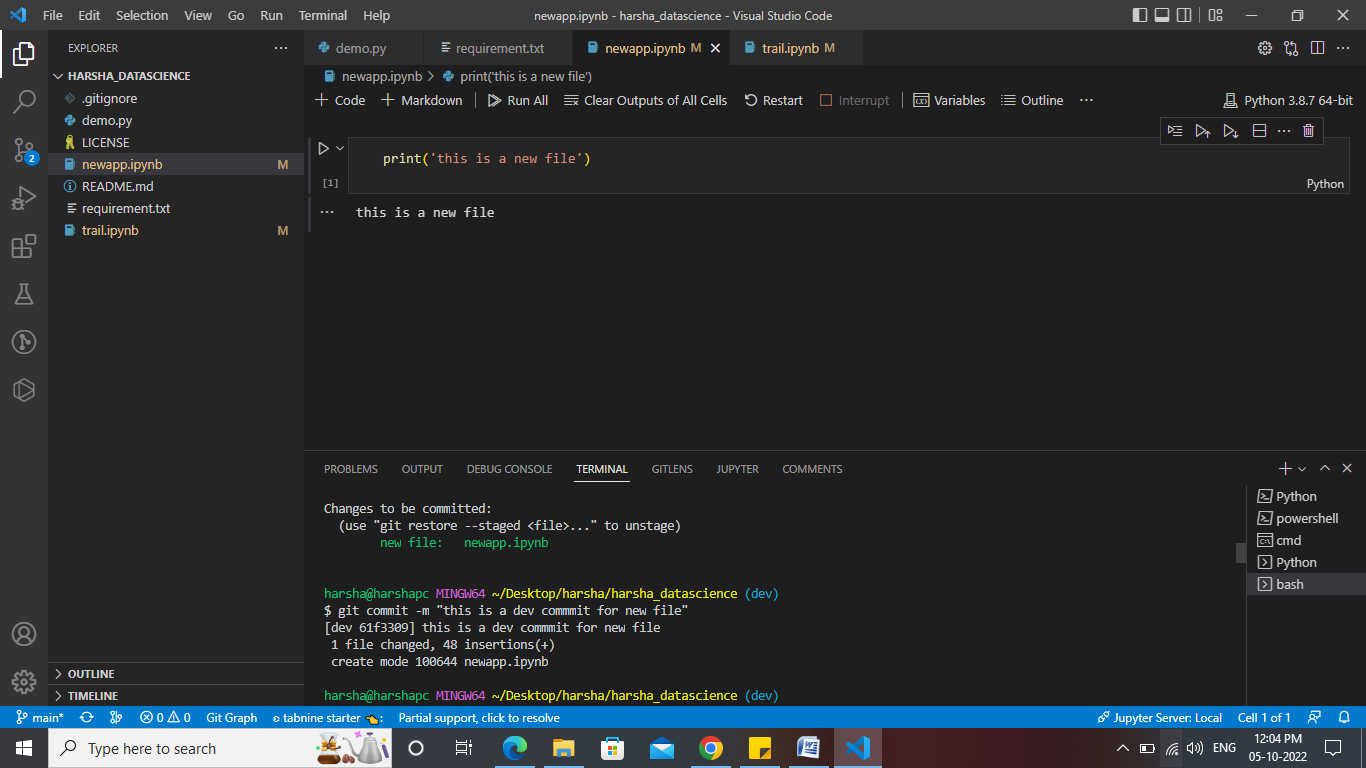
**Screenshot:**

****

**5. git commit -m**

This command commits any files added in the repository with git add and also commits any files you've changed since then.

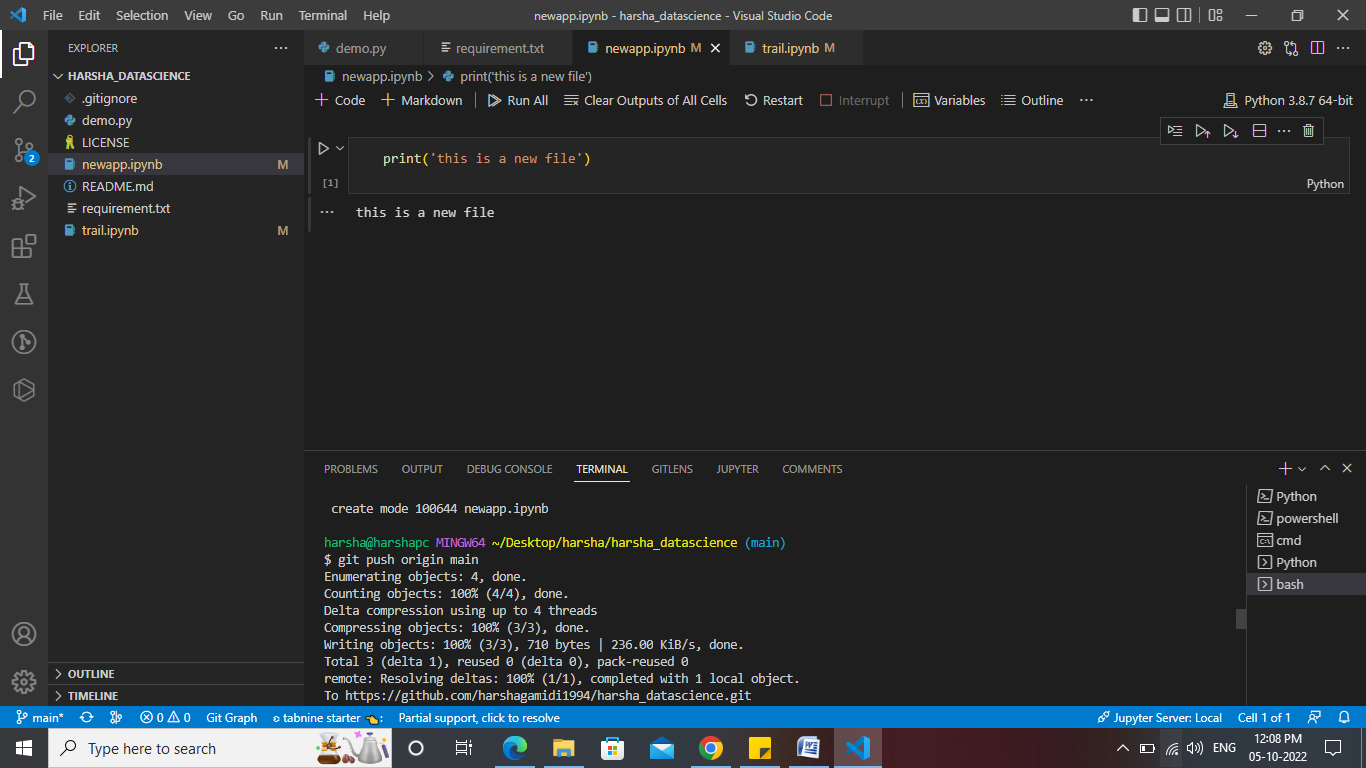
**Screenshot:**

****

**6. git push**

It is used to upload local repository content to a remote repository. Pushing is an act of transfer commits from your local repository to a remote repo.

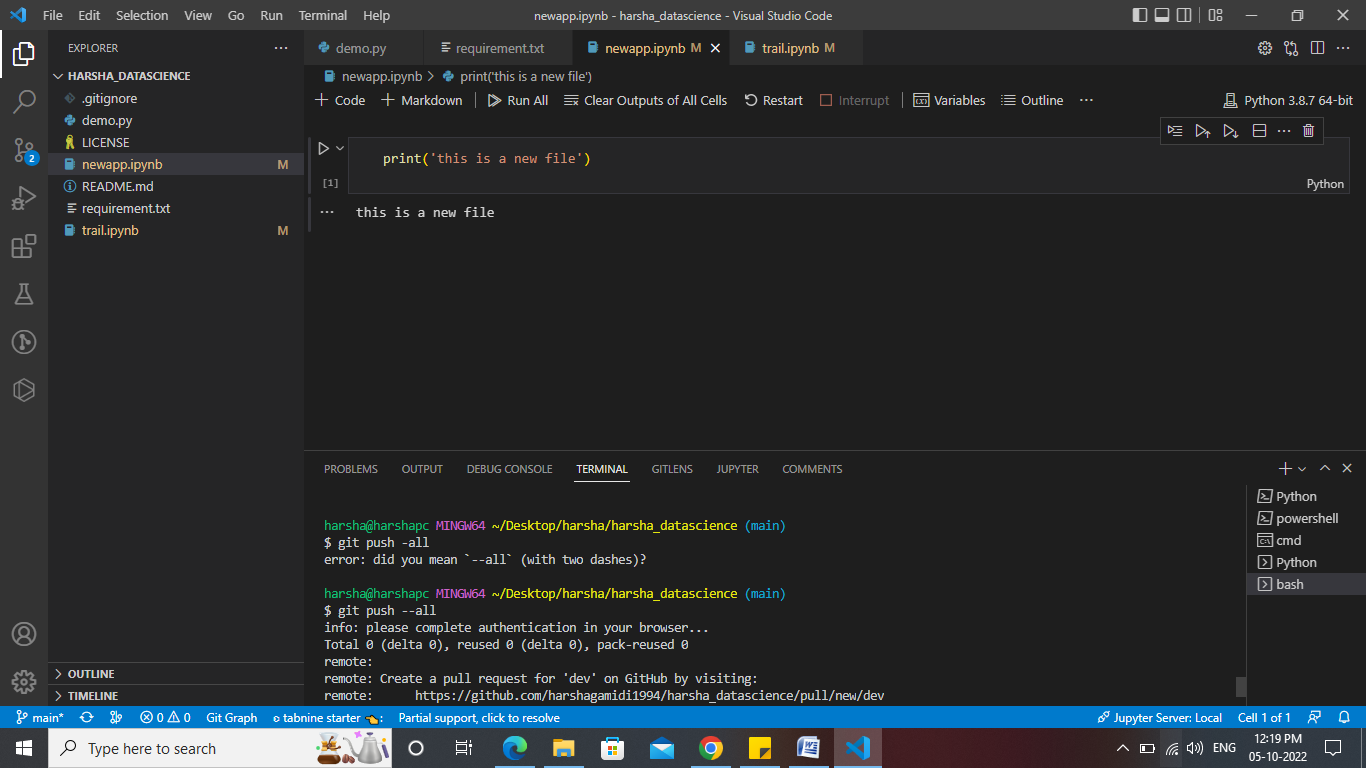
**Screenshot:**

****

**6. git push - - all**

This command pushes all the branches to the main repository.

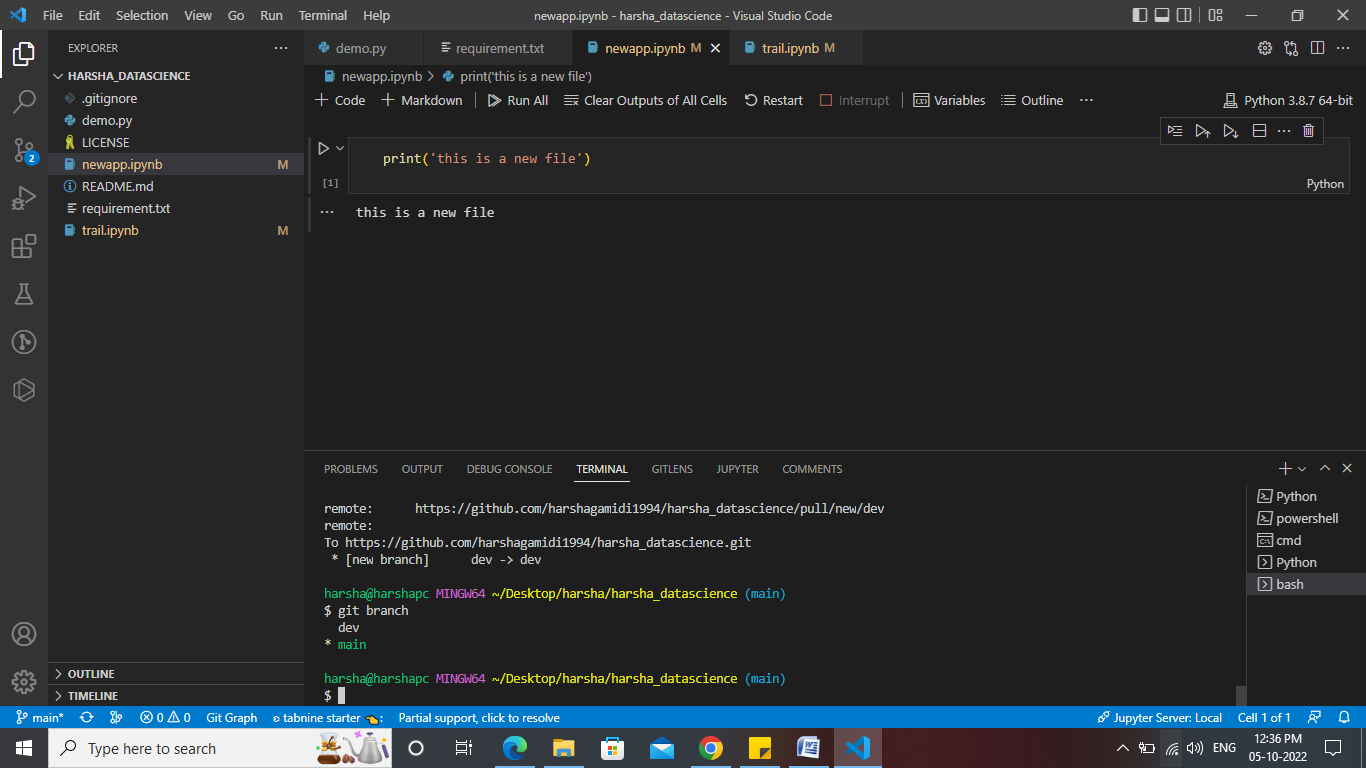
**Screenshot:**

****

**7. git branch**

This command lists all the branches available in the repository.

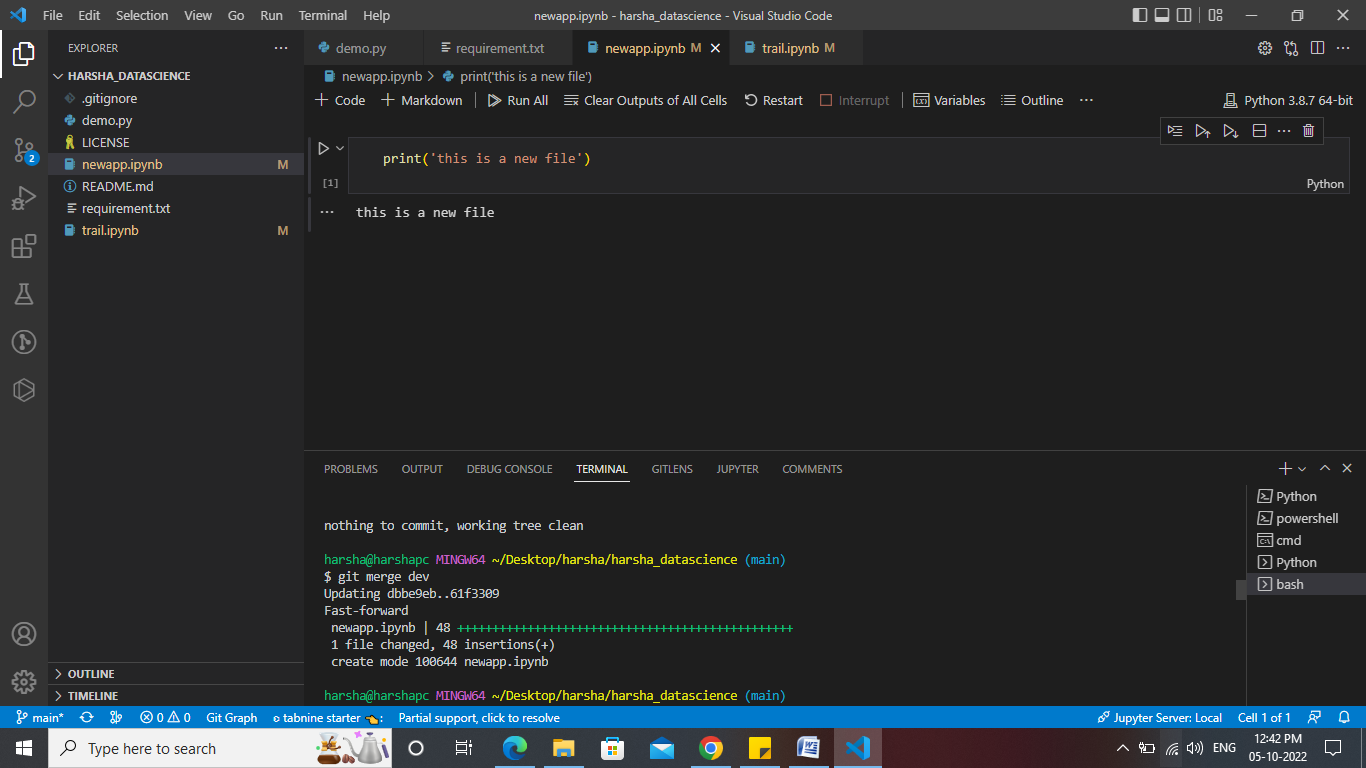
**Screenshot:**

****

**8.git merge Branch Name**

This command is used to merge the specified branch’s history into the current branch.

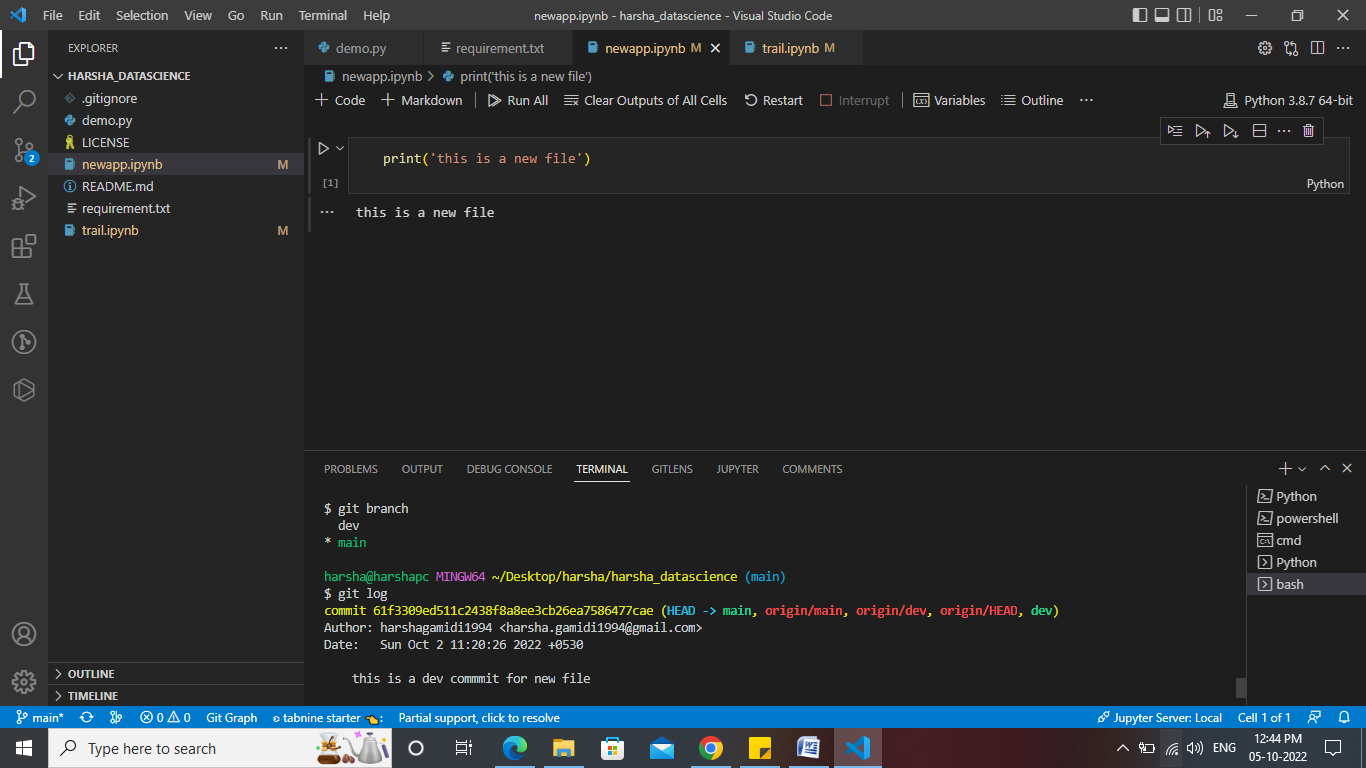
**Screenshot:**

****

**9.git log**

This command is used to check the commit history.

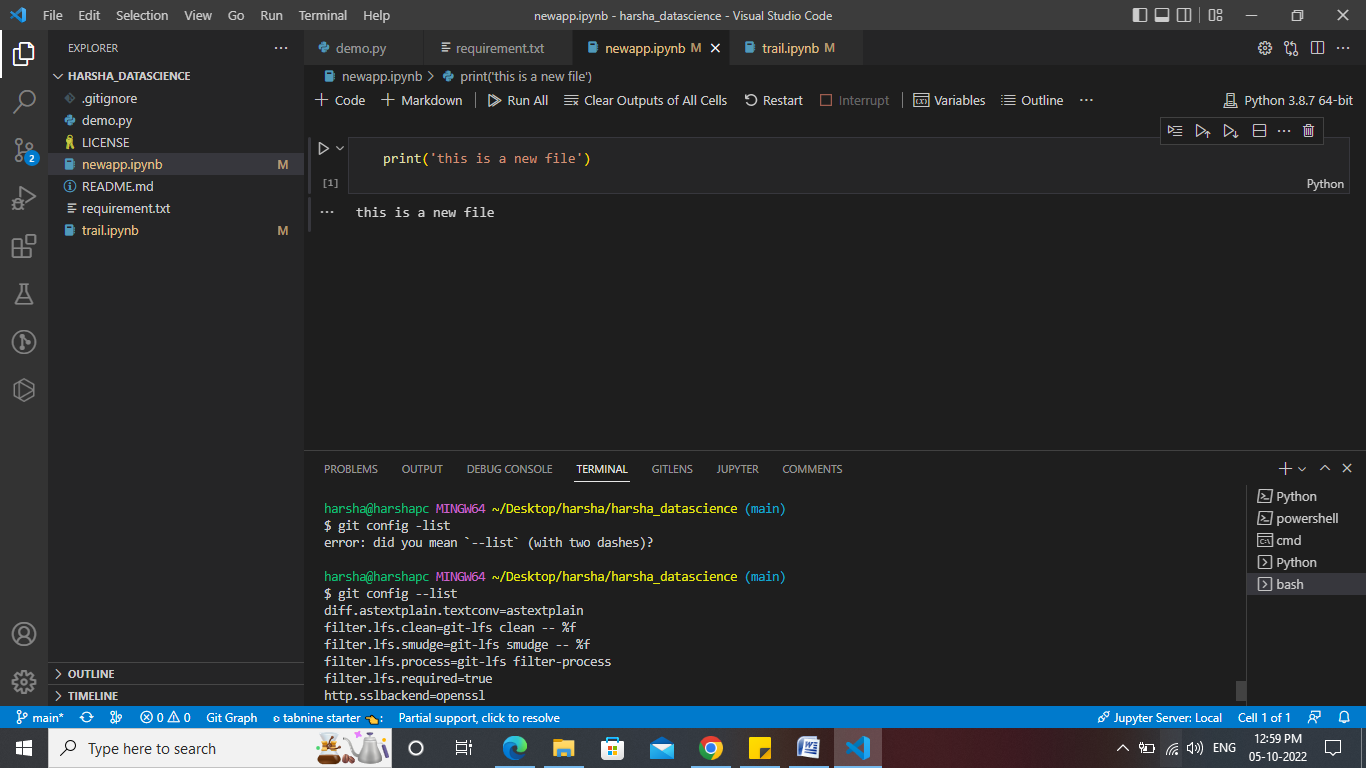
**Screenshot:**

****

**10. git config**

The “git config” command is a function that is used to set and change Git values through three configuration levels: “local”, “global”, and “system”.

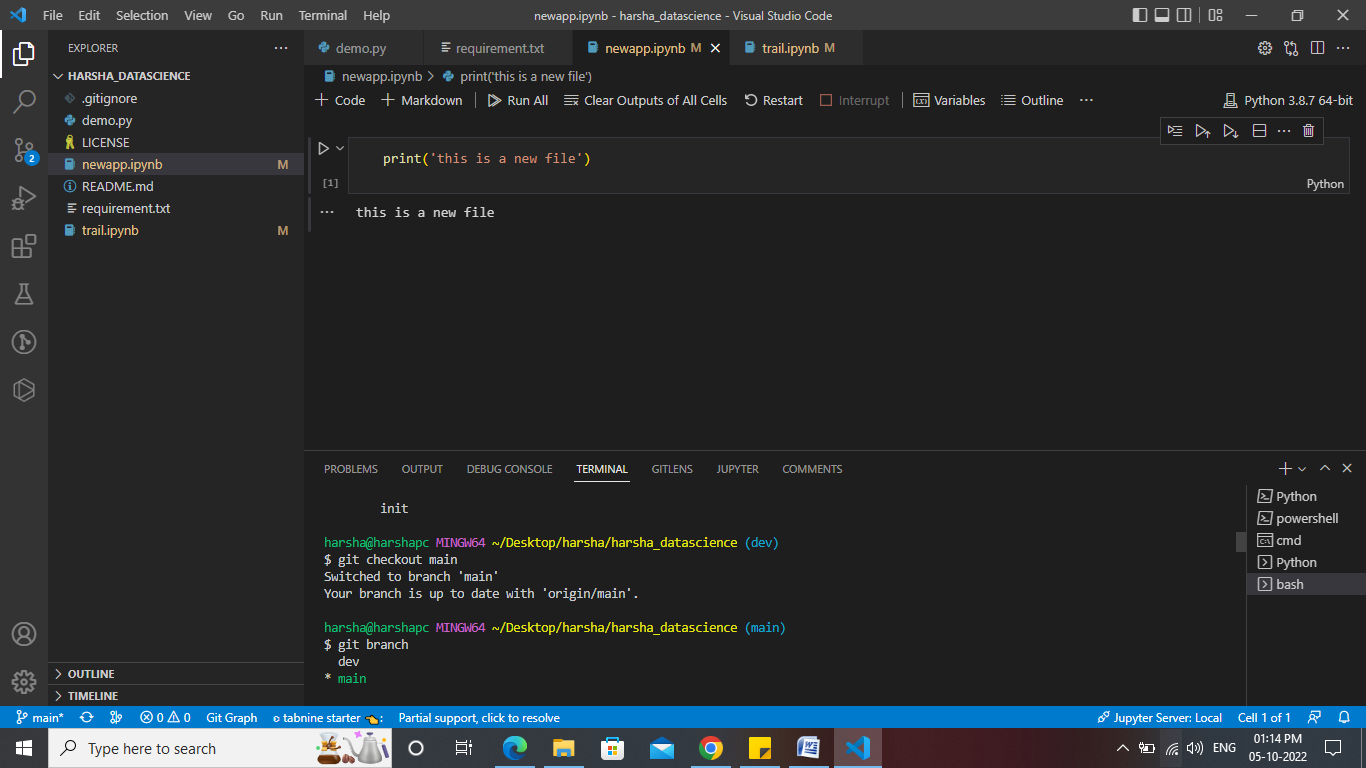
**Screenshot:**



**11.git checkout**

The git checkout command lets you navigate between the branches created by git branch. Checking out a branch updates the files in the working directory to match the version stored in that branch, and it tells Git to record all new commits on that branch.

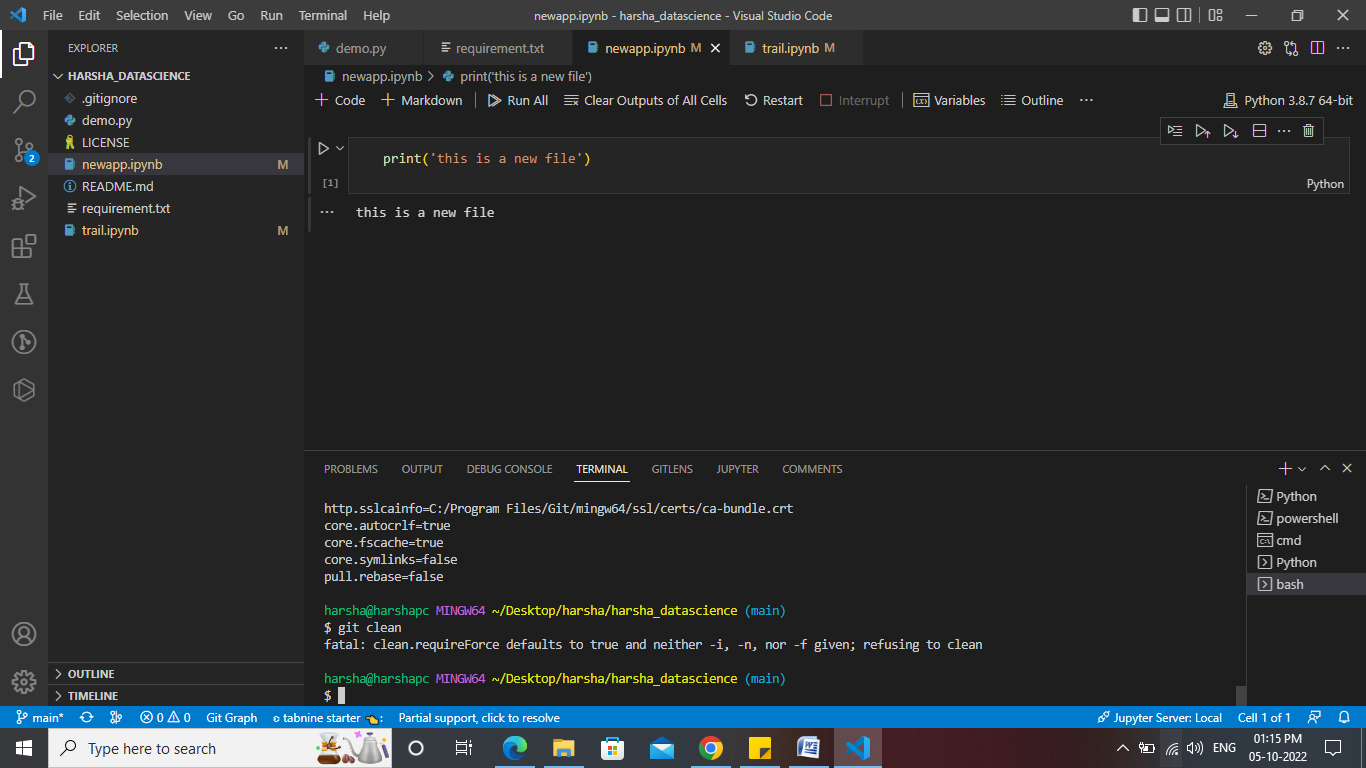
**Screenshot:**

****

**12.git clean**

Removes untracked files from the working directory. This is the logical counterpart to git reset, which (typically) only operates on tracked files.

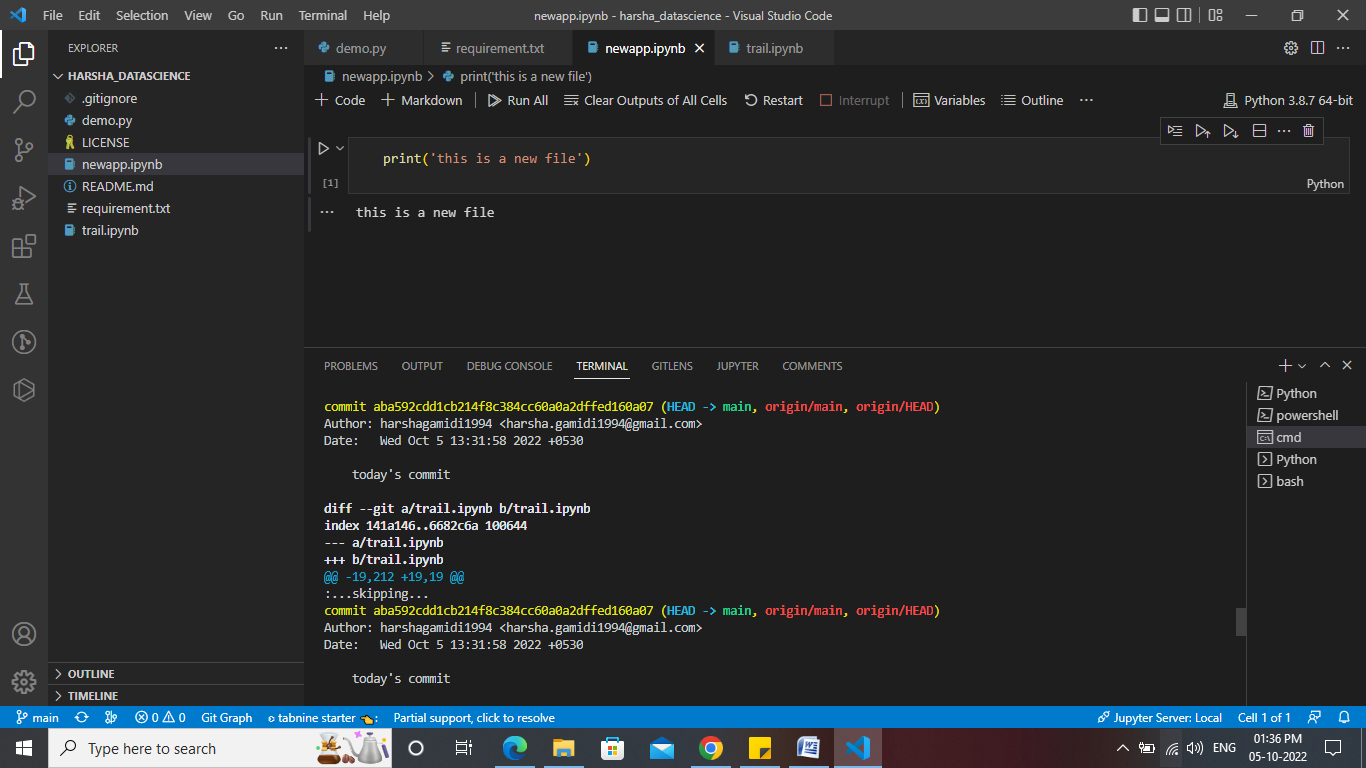
**Screenshot:**

****

**13.git head**

This command will show the location of the Head.

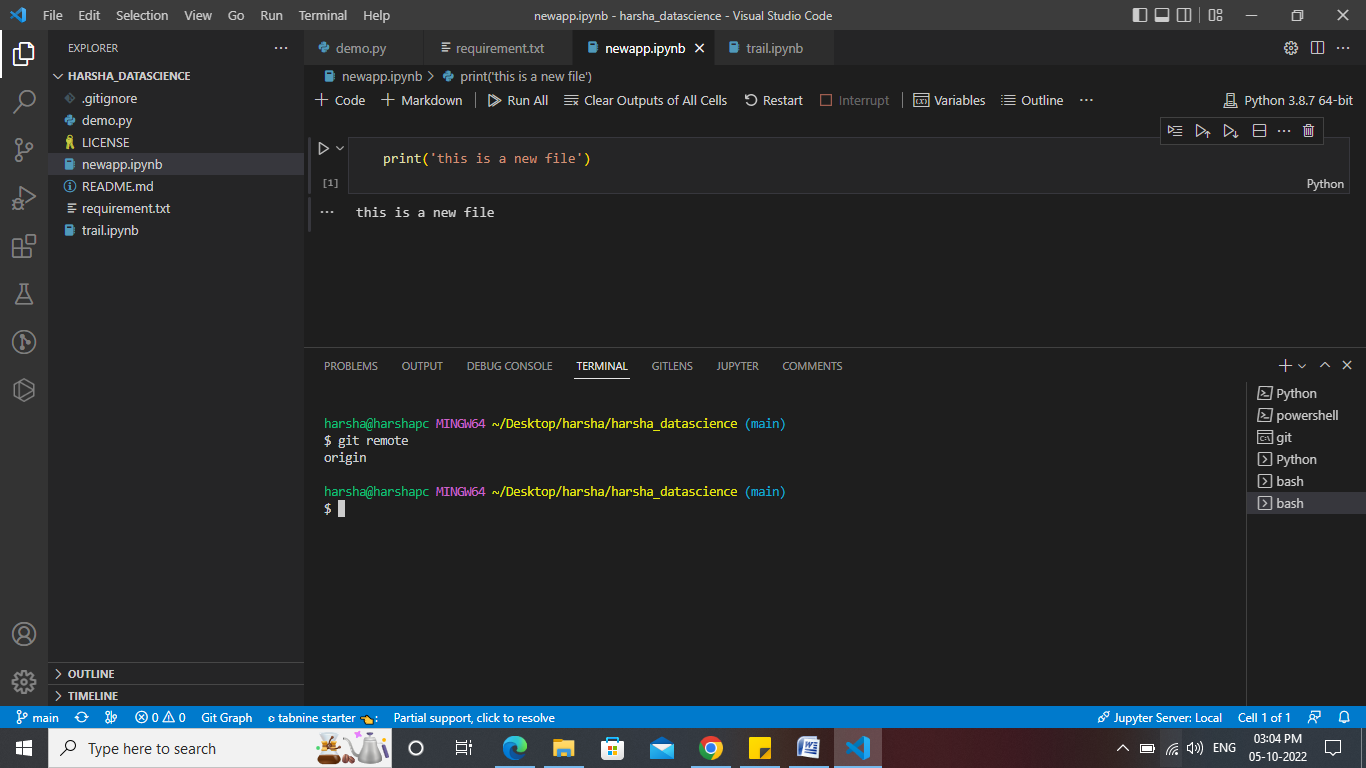
**Screenshot:**

****

**14. git remote**

The given command is providing the remote name as **the origin**. Origin is the default name for the remote server, which is given by Git.

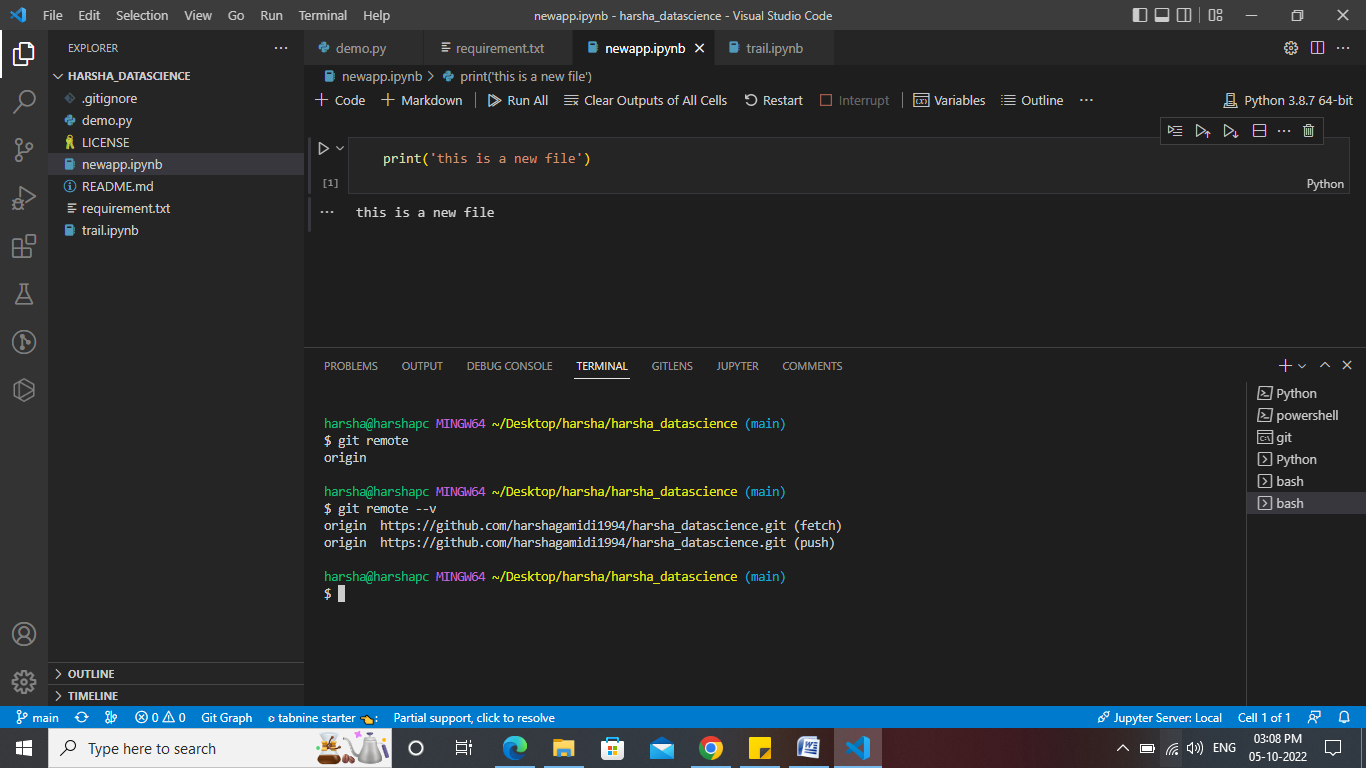
**Screenshot:**

****

**15. git remote --v**

If a repository contains more than one remote connection, this command will list them all.

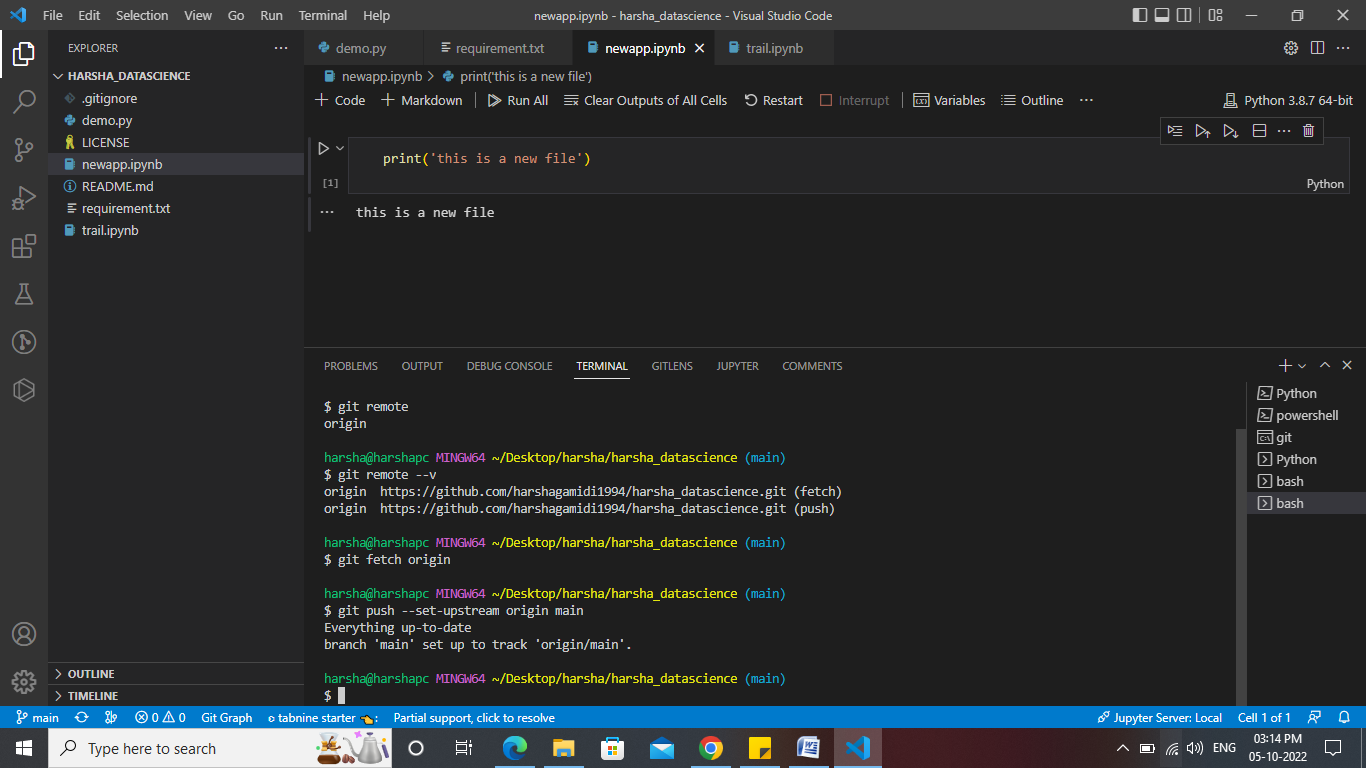
**Screenshot:**

****

**16.** **git push --set-upstream origin master**

This command will set the selected branch as the default remote branch.

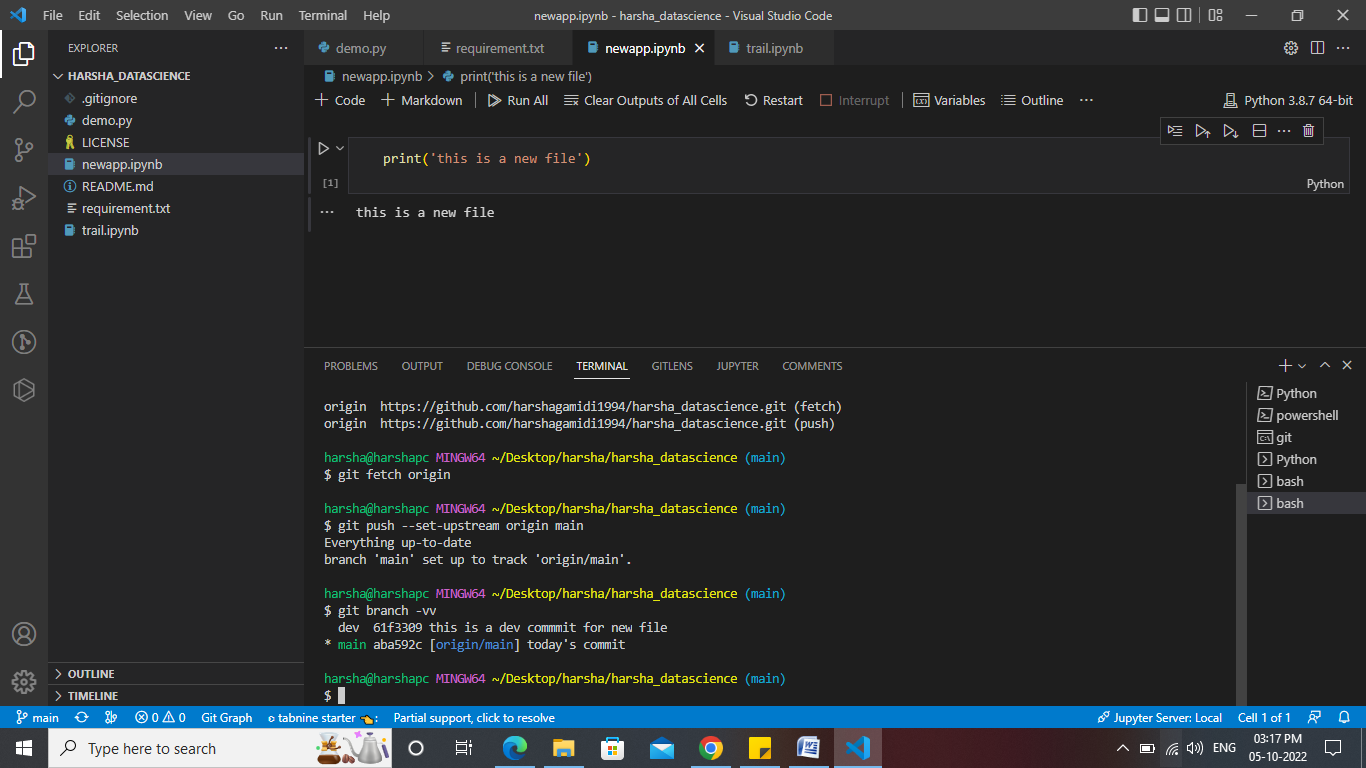
**Screenshot:**

****

**17.git branch –vv**

It will list out the all the branches and will highlight the default branch

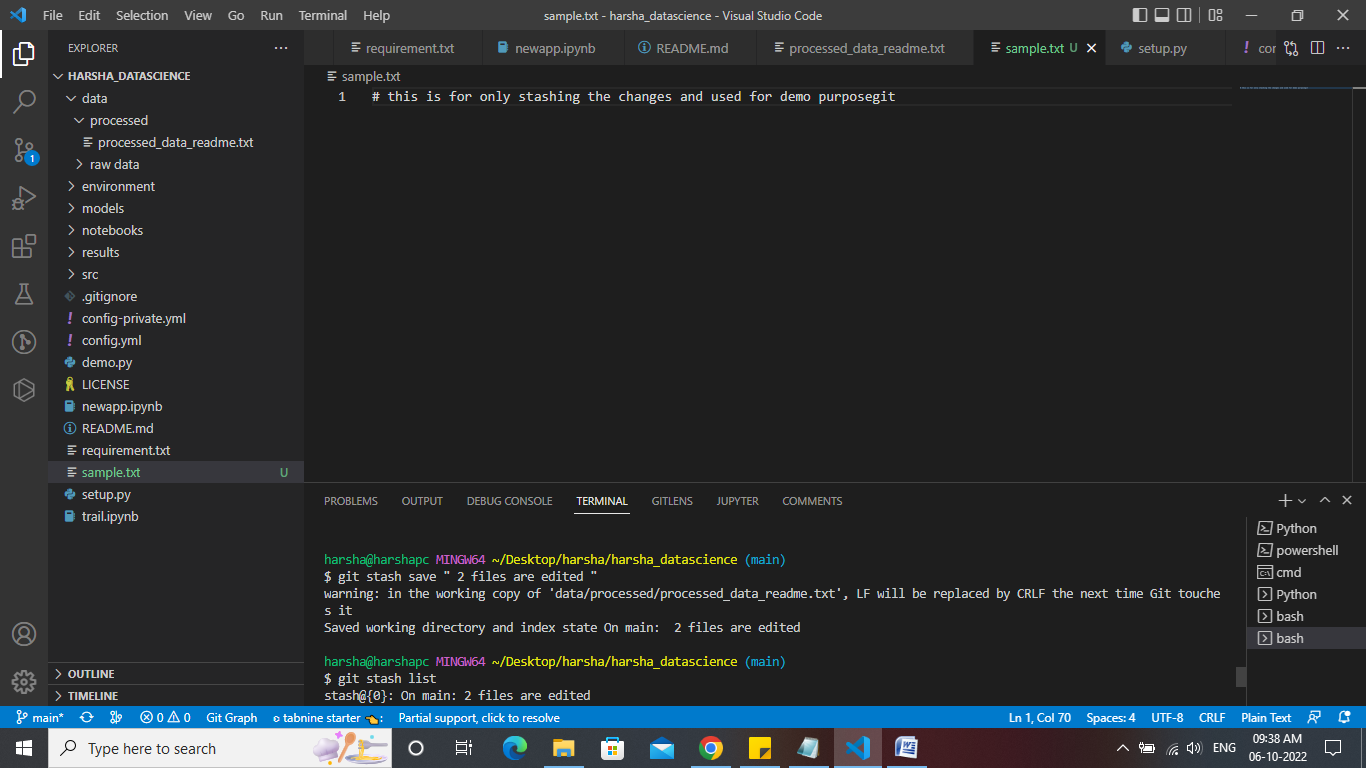
**Screenshot:**



**18. git stash**

This enables you to switch branches without committing the current branch and takes your uncommitted changes (both staged and unstaged), saves them away for later use, and then reverts them from your working copy.

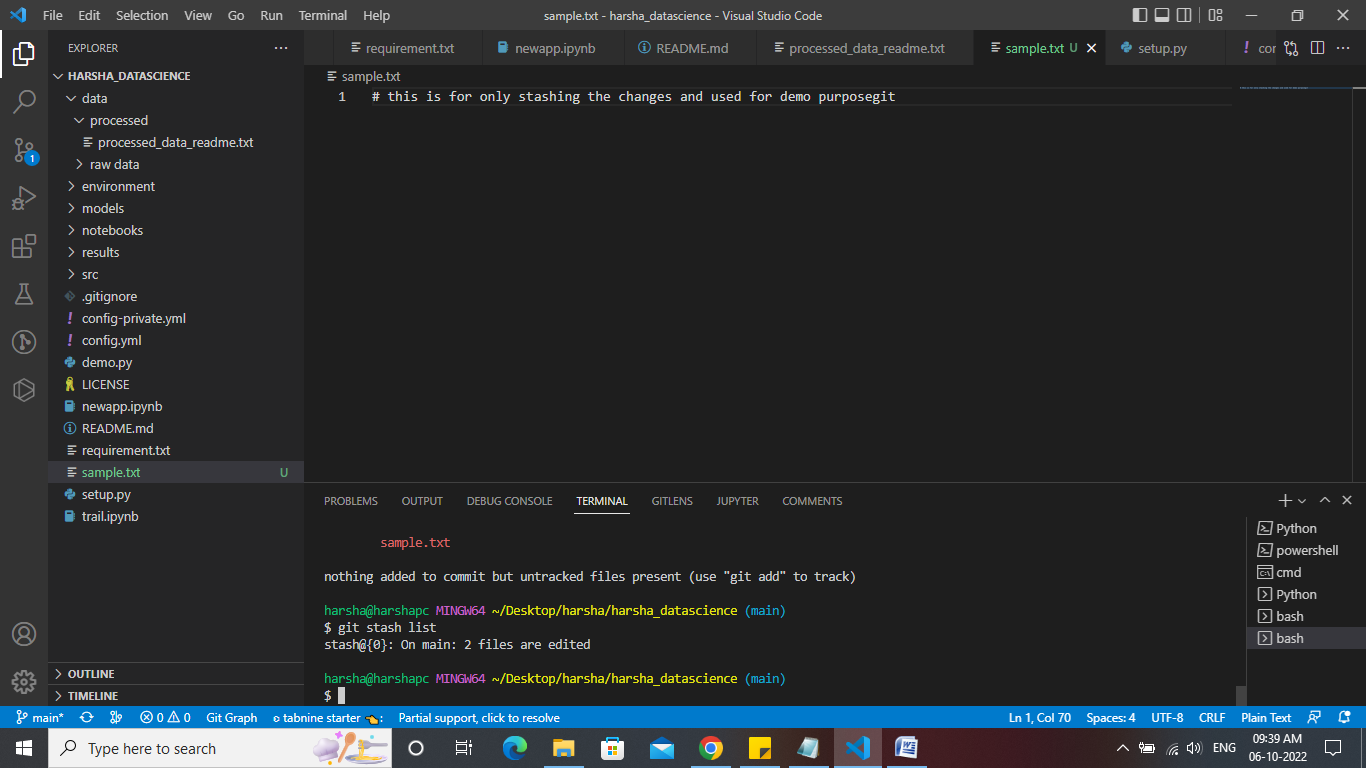
**Screenshot:**

****

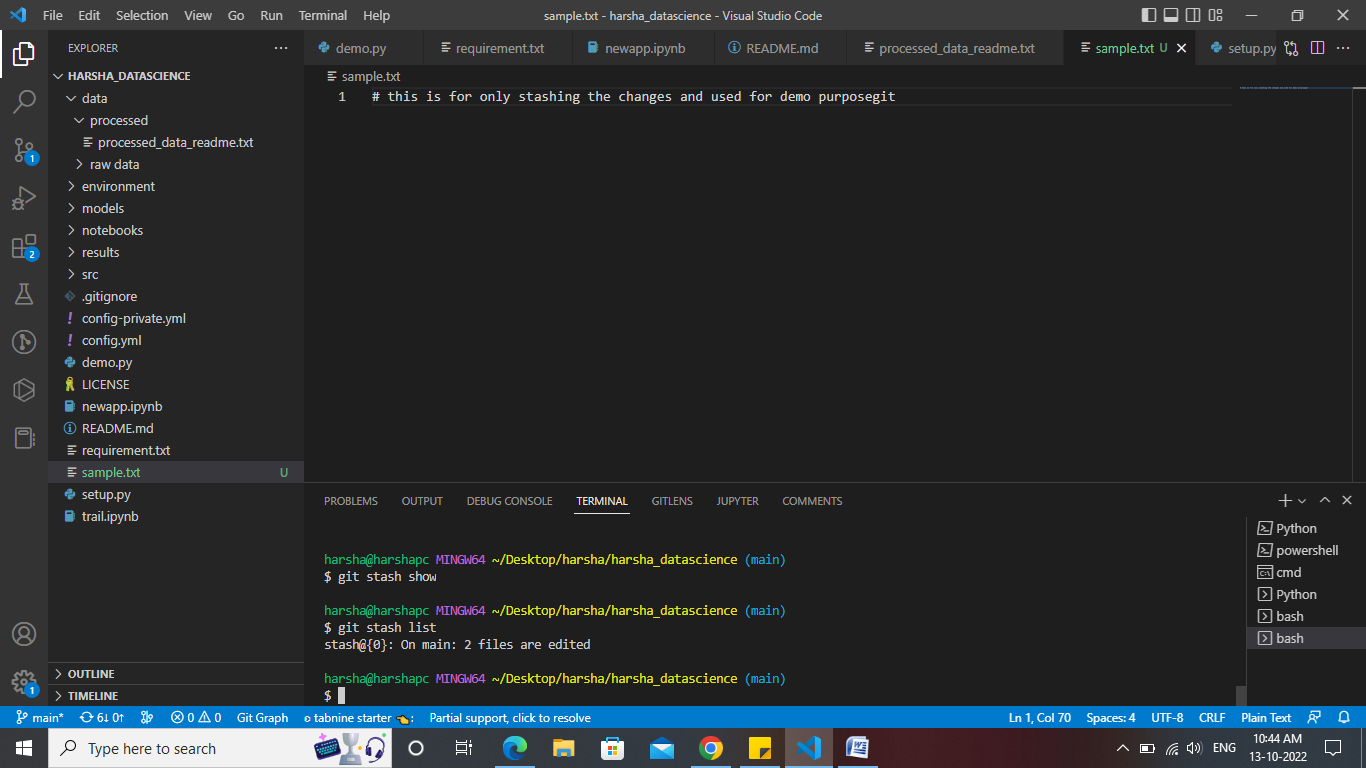
**19. git stash list**

This command helps us to get the list of the stashed files in the below order mentioned in the screenshot

**Screenshot:**

****

**Stash changes**

****