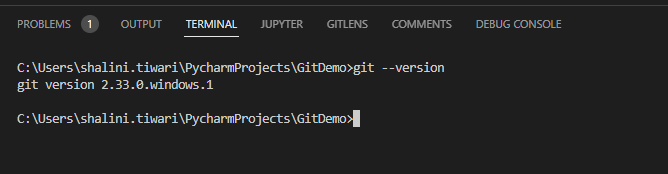
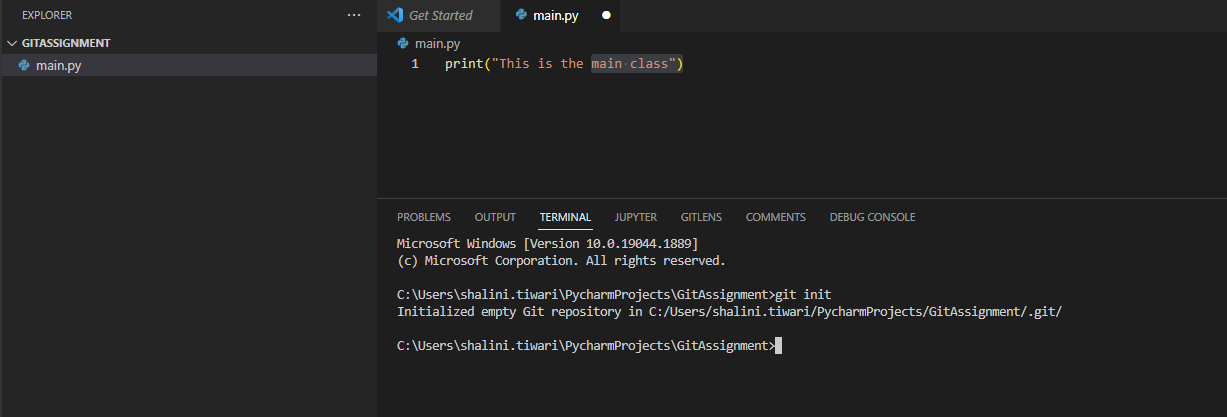
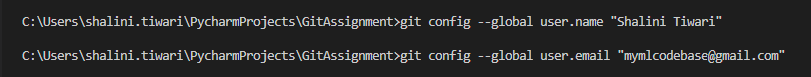
1. **git --version**

This command describes the git version installed in the local system [](https://user-images.githubusercontent.com/114802910/193401158-863c1c90-7ec4-4b6d-a19d-b476200b7c0c.png)

1. **git init**

This command is used to create new blank Git repository. This is the first command that one should run on Git which converts an existing project as Git project [](https://user-images.githubusercontent.com/114802910/193401541-a5e7177f-07a8-4acd-b703-69a831b3f0e6.png)

1. **git config --global user.name “<name of user>”**
2. **git config –global user.email “<user’s email address>”**

These commands help in registering the username and emailid, such that when the user commit their changes in git, it becomes clear which user has committed the changes. It helps when multiple developers are working in same project and are targeting their changes in the same branch.  


1. **git status**

This command gives the status of untracked files in the repository. If in the repository a new file is added or the existing file is modified or a file is deleted, this command tracks the information of all the changes.  
Text

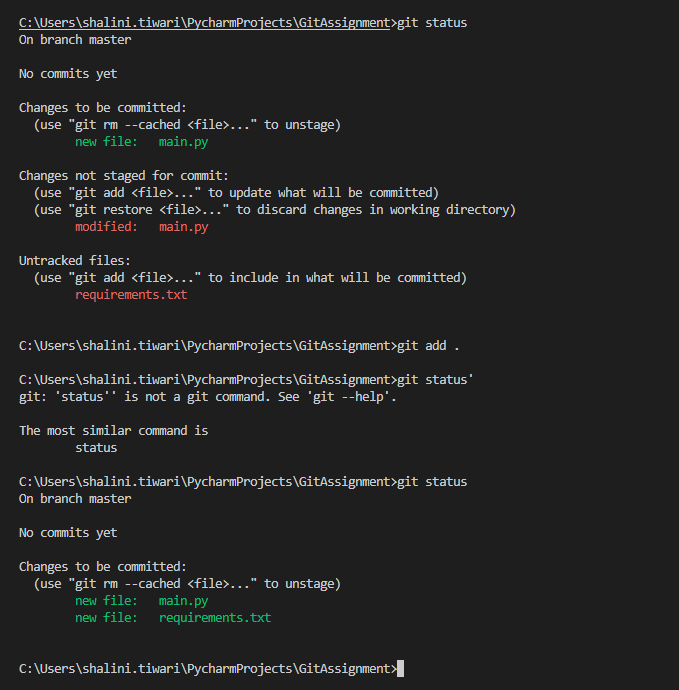
Description automatically generated

1. **git add <file name>**

This command adds the specific file in the working directory to the staging area.  
Text

Description automatically generated

1. **git add .**

This command adds all the changes in the working directory in the staging area. No need to specify the filename.

Note: In git when we commit the code it first goes into stagging environment and when we push the changes, the only it goes into the actual repository.

**Changes pushed in Git repository**

**Git Push**

**Git Commit**

**Git Add**

1. **git commit -m “Check in message"**

This command helps in committing the code/file in stagging environment.  
Text

Description automatically generated

1. **git branch**  
     
   This command let user create, list, rename, and delete branches  
     
   9.1. **git branch** – lists all the branches  
   Text

   Description automatically generated

Note : By default master is the local branch but in git the default branch is main, so before pushing changes in repository, we must change the branch to main using below command:

* + git branch -M main

Text

Description automatically generated-

- if we donot change the branch we will get below error:  
   
 Text

Description automatically generated  
  
 - The above error can be resolved using below command  
 **git pull –rebase origin main**  
 Text

Description automatically generated

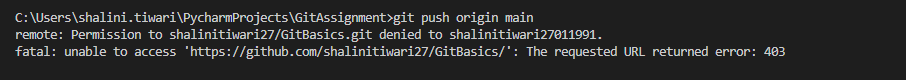
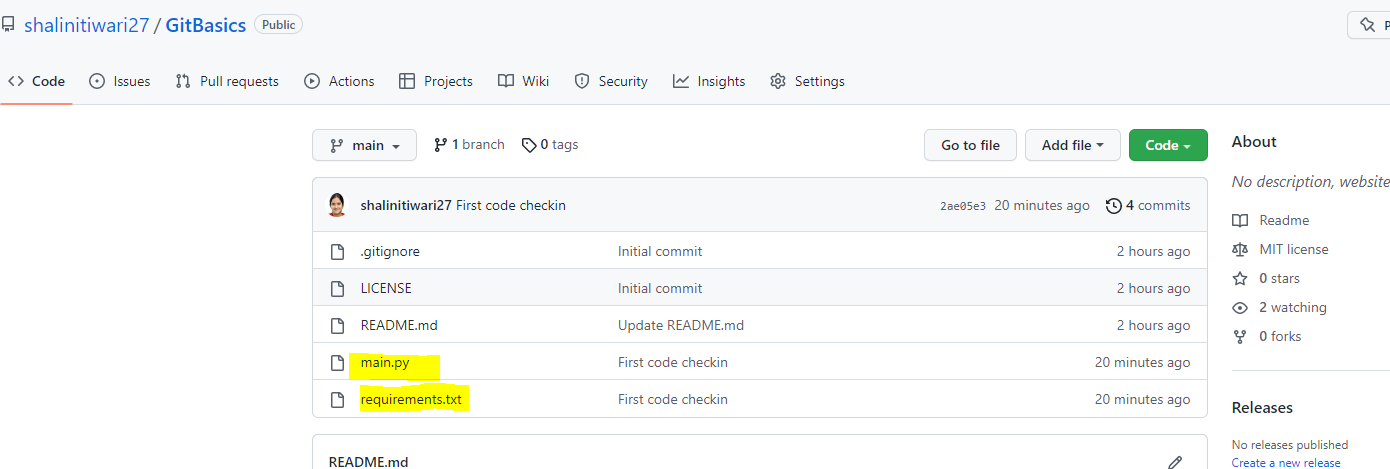
9.2. **git branch -d developer1**   
 Deletes a branch

9.3. **git branch developer1**   
 Create a branch  
Text

Description automatically generated

1. **git remote add origin “<Git repository link>”**  
   This command is used to link the git repository in which user wants to push their local code. It helps in creating connections with other repositories.  
   
2. **git remote -**v  
   This command lists the remote connection user has to other repositories.  
   Text

   Description automatically generated
3. **git push origin main**  
   This command pushes the changes from stagging to main git repository  
   Text

   Description automatically generated  
     
   **Note:** While pushing the changes in my new repository I was getting below error:  
     
     
   Resolution : <https://docs.github.com/en/authentication/troubleshooting-ssh/error-permission-to-userrepo-denied-to-other-user>  
   I added the other repository as my collaborator because both my repos were using the same key. After doing it I was able to push my code in the main repository.  
   
4. **git checkout <branchname>**  
    This command makes the mentioned branch as the focus branch in which developer can make the changes. For example: We have default main branch and one new branch developer 1 in which all the dev changes should go. So by default user is always on the main branch. In order to switch to the developer branch one can, use this command.  
   Text

   Description automatically generated
5. **git merge developer1**  
   This command is used to merge developer1 branch changes in main.
6. **git log**

This shows the sequence of events performed in the repository  
Text

Description automatically generated

16 **git revert <commitid>**  
this command is used to revert the changes pushed in the repository.