DevOps Lab

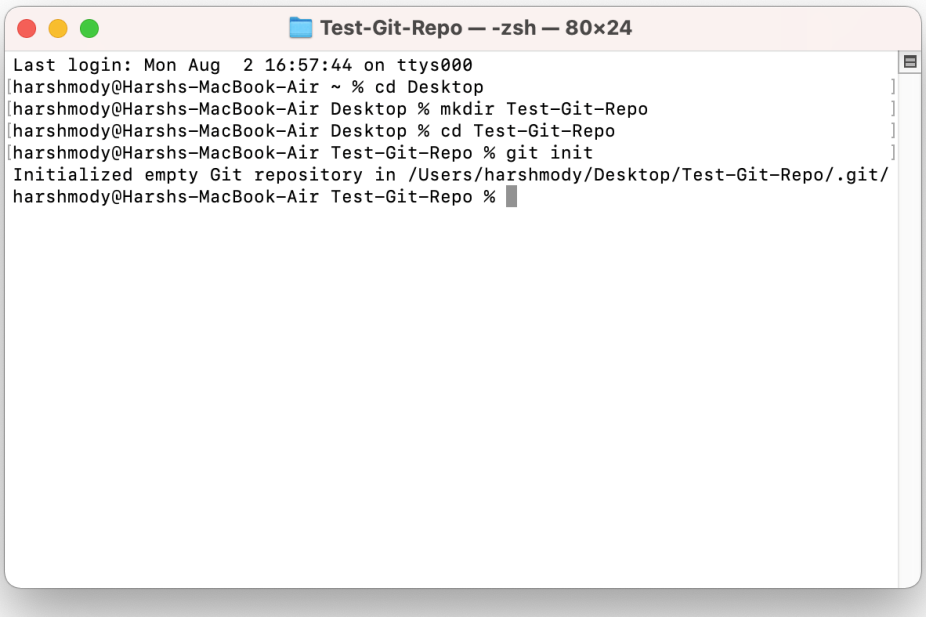
Assignment 3:

Aim: To understand various commands in Git and use GitHub to create a remote copy of our local code.

Commands & Their Execution:

Creating a local repository:

i) Initialize the repository using the **git init** command.



ii) The **git status** command is used to know the status of files in the current working directory.

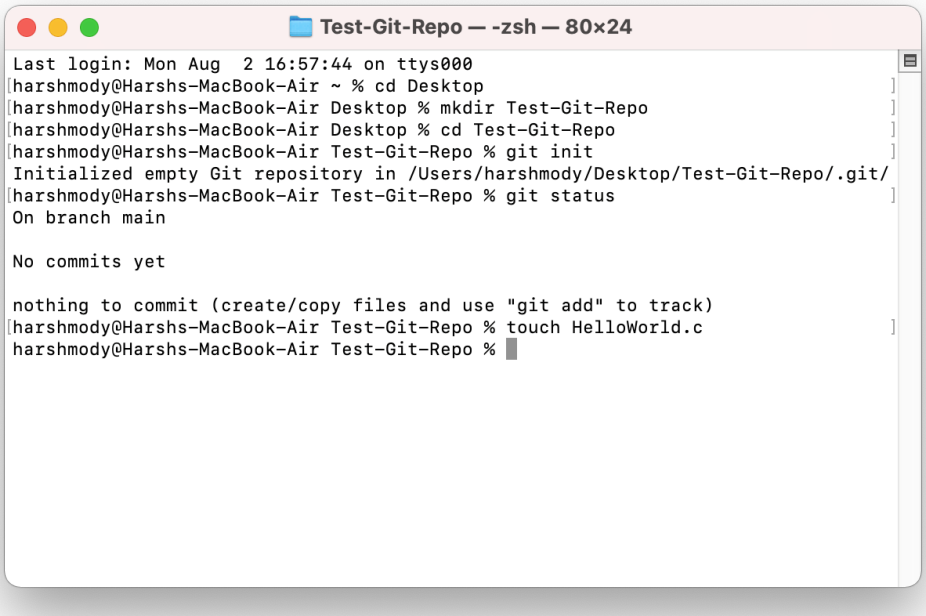
There are 4 basic types of files in any repository which git status command tells us about:

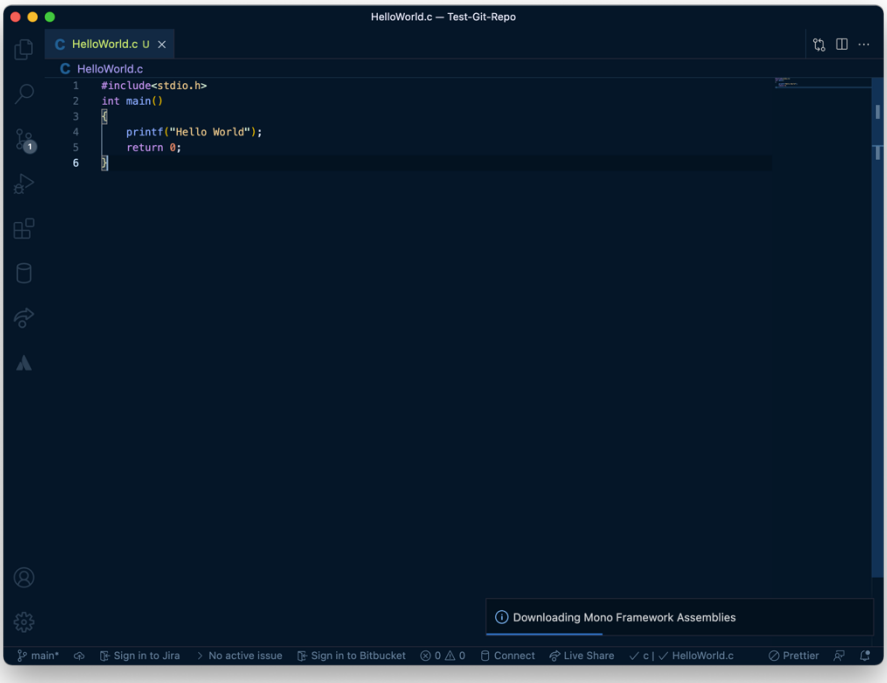
a) Untracked Files: These are newly created files in the repository folder which are not yet added to the repository.
b) Added Files: All files which are not ignored by git using .gitignore file are a part of this group. These are called the staged files.

c) Modified Files: These are files which are already added to git repository but have been modified since and are so are not in the latest version of the repository.

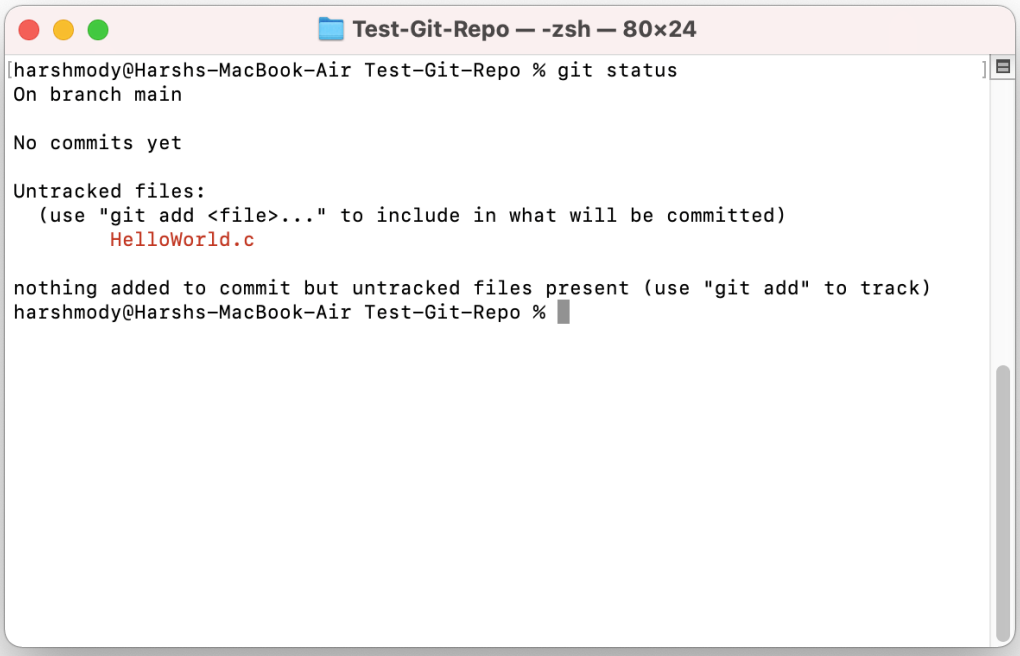
d) .gitignore: This file usually tell repository to ignore files of a certain pattern. For e.g. JetBrains IDE often create .idea files, etc. which do not need to be part of our code repository.

Using touch command we create a new File and Using VS Code, we add our code to the file



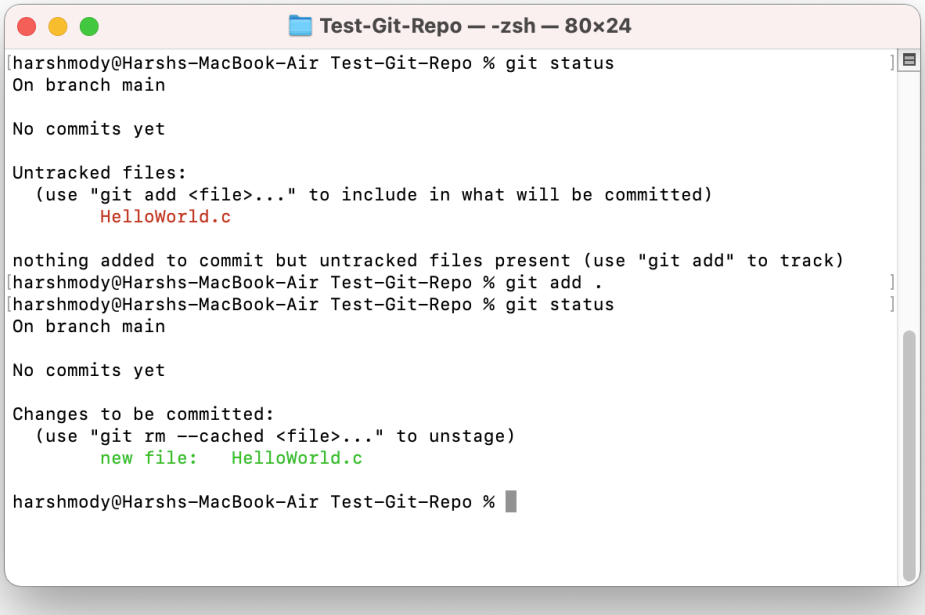


Now, since we created the file but we didn't add it to our repository, git status shows that's it's a untracked file.

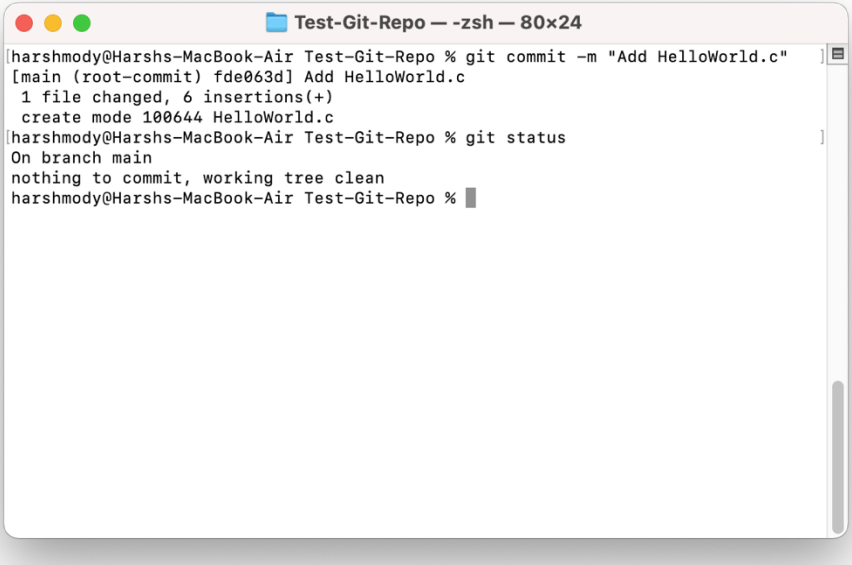


iii) The **git add** command is used to add files in current working directory to our repository.

The **git add .** command adds all the untracked files in current working directory to the repository.

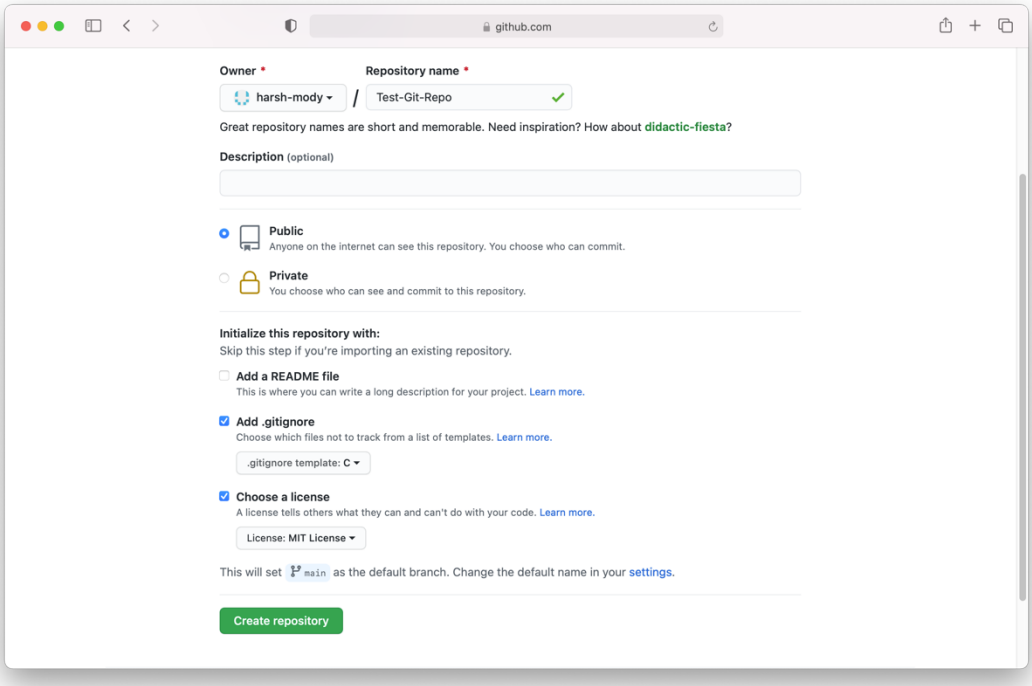


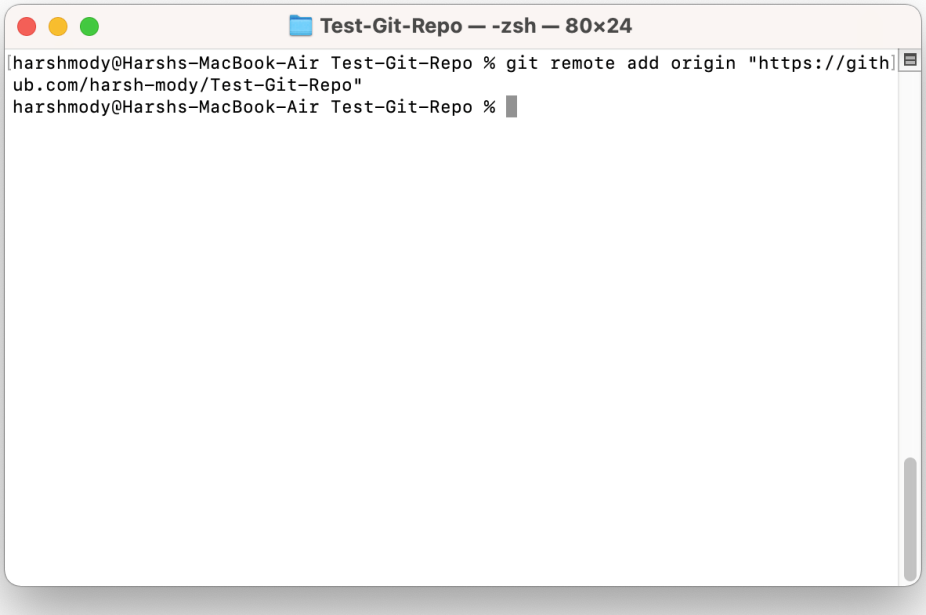
iv) Now, our added files are in the staging area and to make these changes a permanent checkpoint to where we can revert our repository in case we find bugs in latest version of our code, we use the **git commit** command.



As we see, after committing changes, our working tree is clean and now we have a checkpoint where we can revert to in-case we need to.

v) Now, adding these changes to a remote hosted repository on GitHub, we create a new repository.



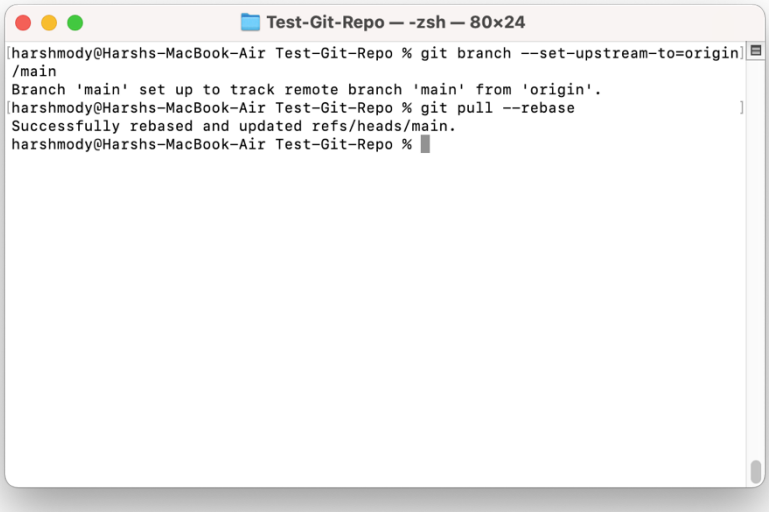


vi) The **git pull** command is used to pull the changes from the remote repository.

But, before we push, we need to pull changes from initial commit on remote. So, we need to use **git pull** command. But before that we need to set a default

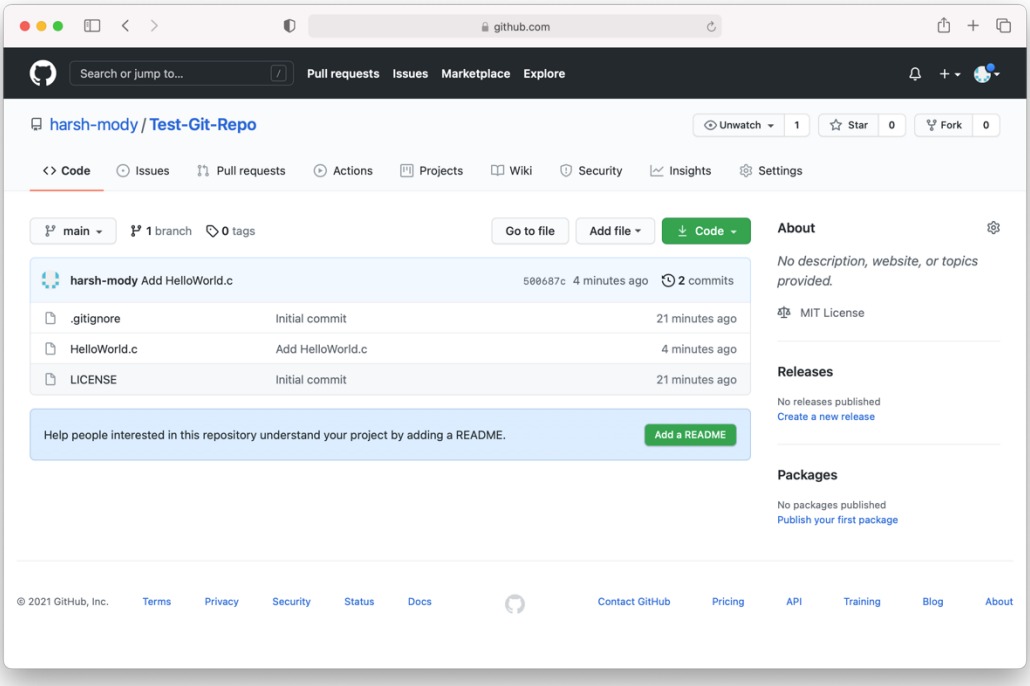
branch to pull from remote repository. So, we use **the git branch --set-upstreamto=origin/main** command.

Here, since I had created an initial commit on remote repository, and the local as well but the version history of the two repository's did not match, we use the **git pull --rebase** command.

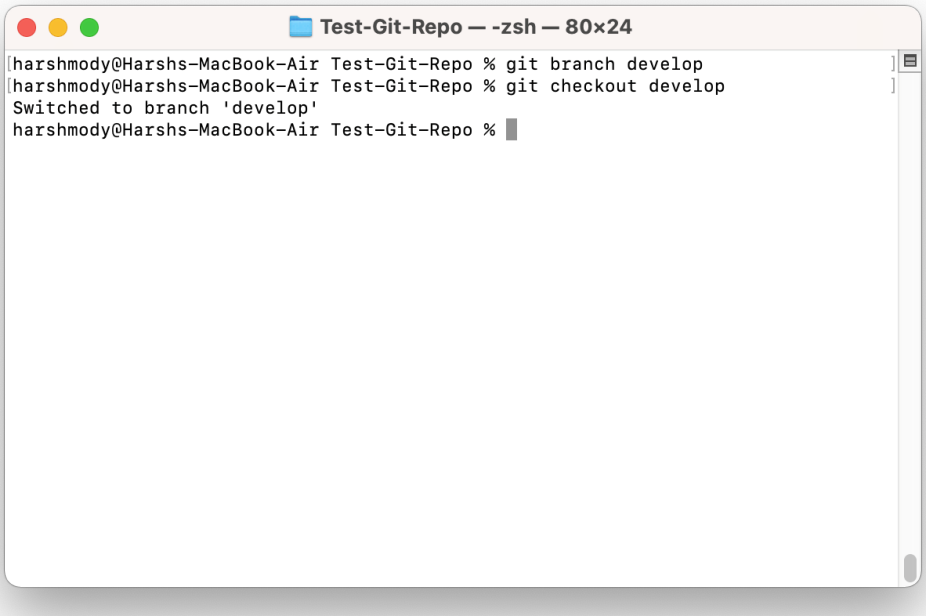


vii) The **git push** command is used to push the changes from local repository to remote repository

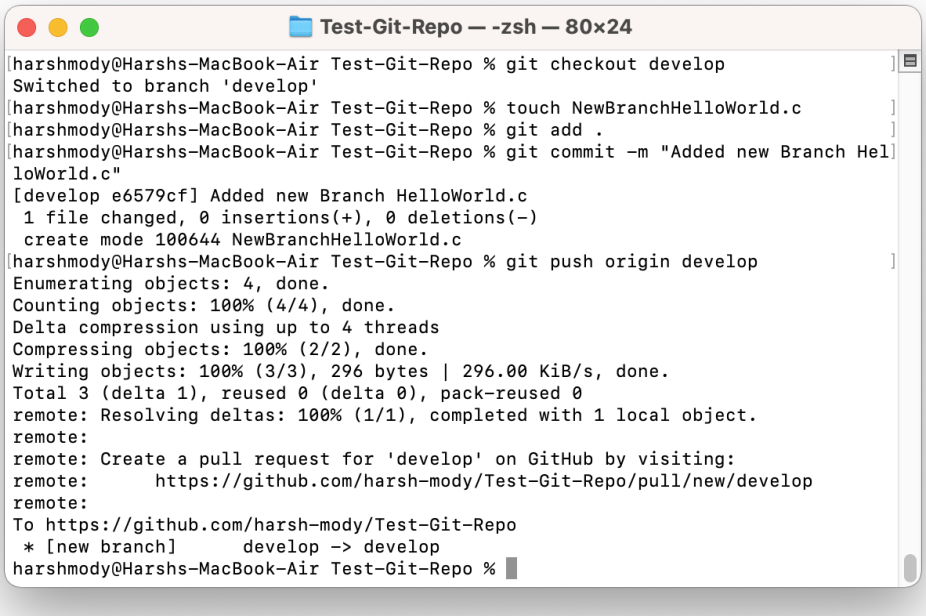
As we can see, our HelloWorld.c file was committed to our remote GitHub repository.

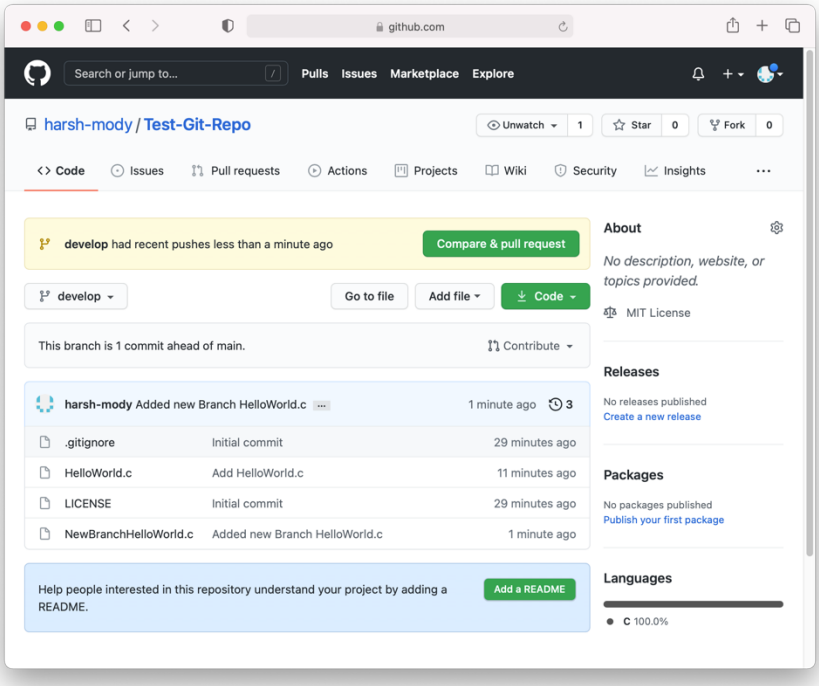


viii) Now, creating a new branch using **git branch develop**, we can create a new branch. The **git checkout develop** command is used to navigate to the newly created branch.



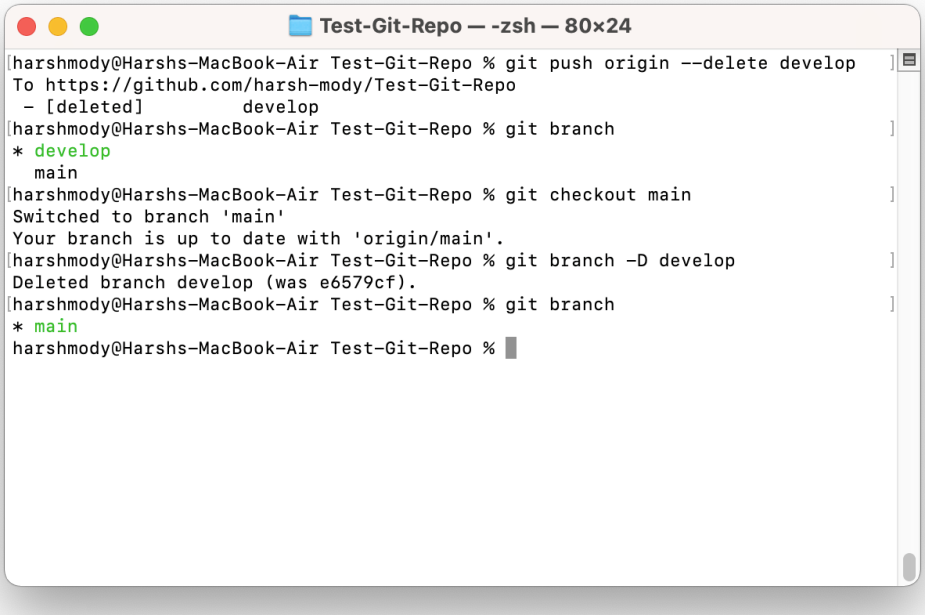
Using similar add, push commands, we create a new file and push it to new branch in remote repository. Thus getting our new branch to remote repository as shown below.

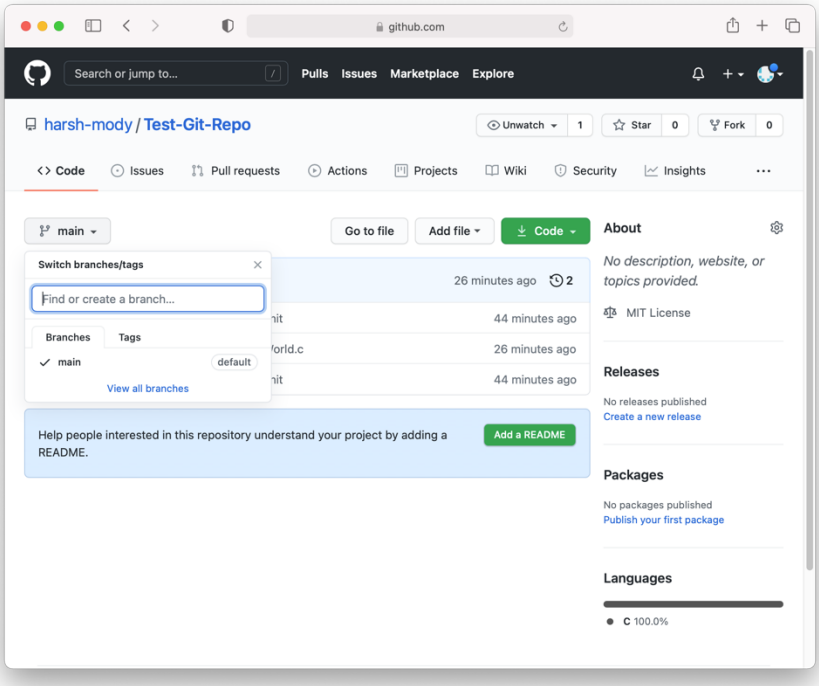




Now, we delete the created branch and push these new changes to remote repository.

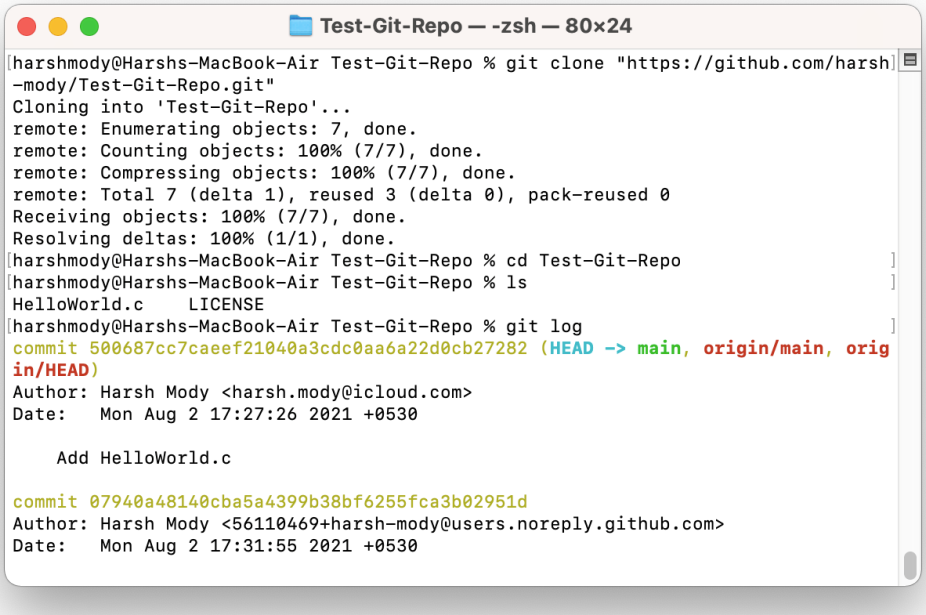
To delete develop from both remote and local repository, use **git push origin -delete develop** to delete it from remote and then **git checkout main** to checkout of main and then **git branch -D develop** to delete the local copy of the branch.





ix) The **git clone** command. This command is used to pull/bring latest changes from a remote repository to local repository we just created on GitHub using the command **git clone "https://github.com/harsh-mody/Test-Git-Repo.git**.

As we see, all the changes added to remote repository were pulled correctly.



Conclusion: Thus, successfully understood various basic commands in Version Control Systems Like Git and also understood the use of remote repository hosting services like GitHub .