**GitHub**

GitHub is a devolopment platform for open source to business,we can host code and review code,manage projects and to build software alongside millions of other devolopers.Anyone can contribute to the linus kernal. In this world millions of software programmers they can be work on single project how can they manage is it easy to manage that, no its very difficult to manage that project .In Git merging of all these sources merging of different things, different people provided by Git .

How Git Works?

Git is actually a distributed version control system,and we have another concept SCM(source code management) or (software configuration management), if you are working on a project or if you are alone to working on a project , when you get requirement from someone, what you are doing is you make one product a small product like MVP(Minimum Value Product) for that you have acode & features you devoloped. After some days you realised that if you add some more features on that and those new features were not working properly what you will do you can remove that features.After some days your client says earlier you changed those features now we want those features they were really good, then what will you do luckily if you have backup files on your drop box and copy that files and use it. If you have lot of work after that you can changes that features & remove to maintain that versions for that we have a version control system,when you released a first version that is snapshot version, after that 1.0,1.1,1.2…2.0 all changes were created as one text files and Readme files on our folder if you imagine if you maintain all those files & which feature is which version why you have to maintain all those thingswe have a powerful tool for that i.e, Version Control System. We have no.of websites like GitLab,GitHub,BigBucket…etc. Earlier they were used (Centralised Version Control System) CSVS, now Distributed Version Control System(DVCS).

In centralised version control system if you have one system and if you have one server and all versions are stored on that server, if you add new features previous versions are in server.Here we have a drawback what it is you have a copy of code & Versions is in server. If your server is down or no internet connection that time how will you get information from your server repository. If you have a multiple systems version is stored in server and every system can have a local copy of repositories ,if you have no internet connection that will be fine why because we have a local copy we can use that this is called as Distributing the repositories, here Git will use distributed version control system.Git will help you to handle those things if we save and commit ,push those things .this is what Git will works.

Another Concept as Trunk Based Devolopment:

Here if you devoloping one project the current version is 4.7 & if you have another feature adding that on 4.7 version that will be on 5.0.

If you working on 5.0 and old one is 4.7, somebody says there is a bug on your 5.0 ,if we have availability to run simultaneously on two versions.

Trunk Based Devoloped:

Master Branch

4.7

4.7 adding feature & save

4.7 adding feature & save

Here we can commit & save and pushing feature 5.0 New Branch

4.7 parallel used two versions at a time.

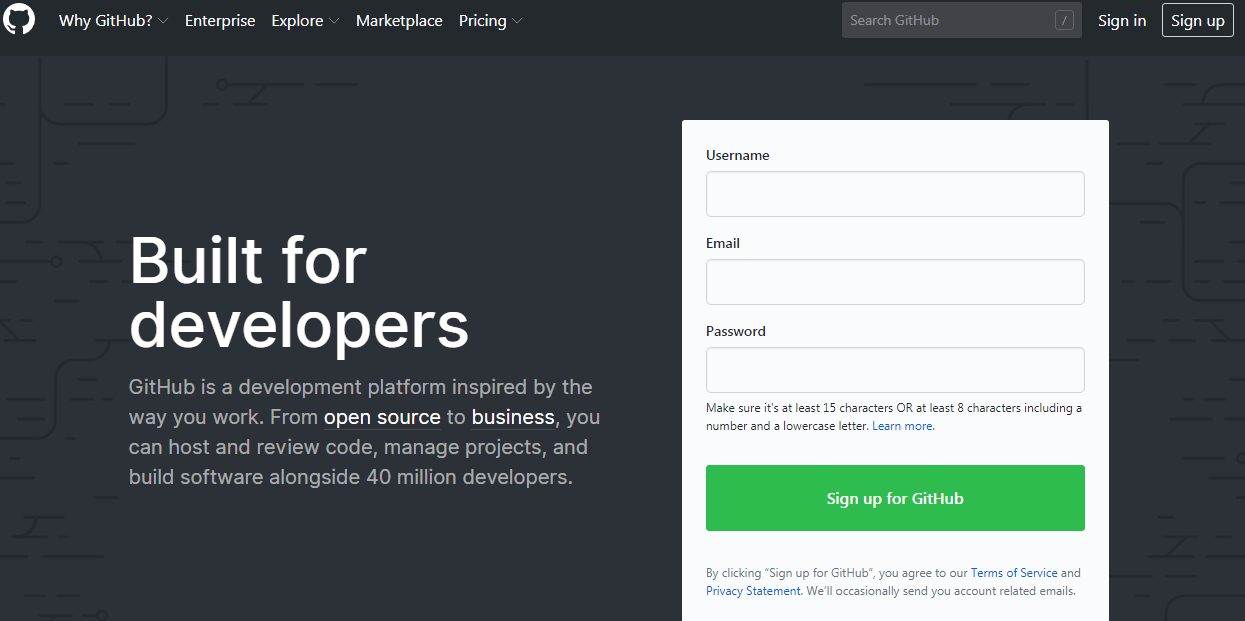
If you are using 5.0 then that time you lose the older version of 4.7 why because you save and commit pushing new features on the older version.

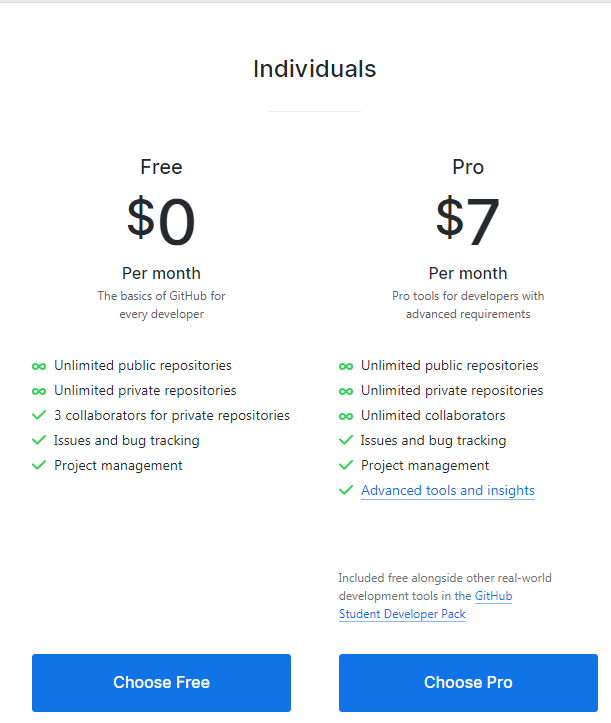
Don’t’ work on the same branch, if you want to work on 5.0 versions then you create another branch that is possible on Git & you can still continue on 4.7 also simultaneously.

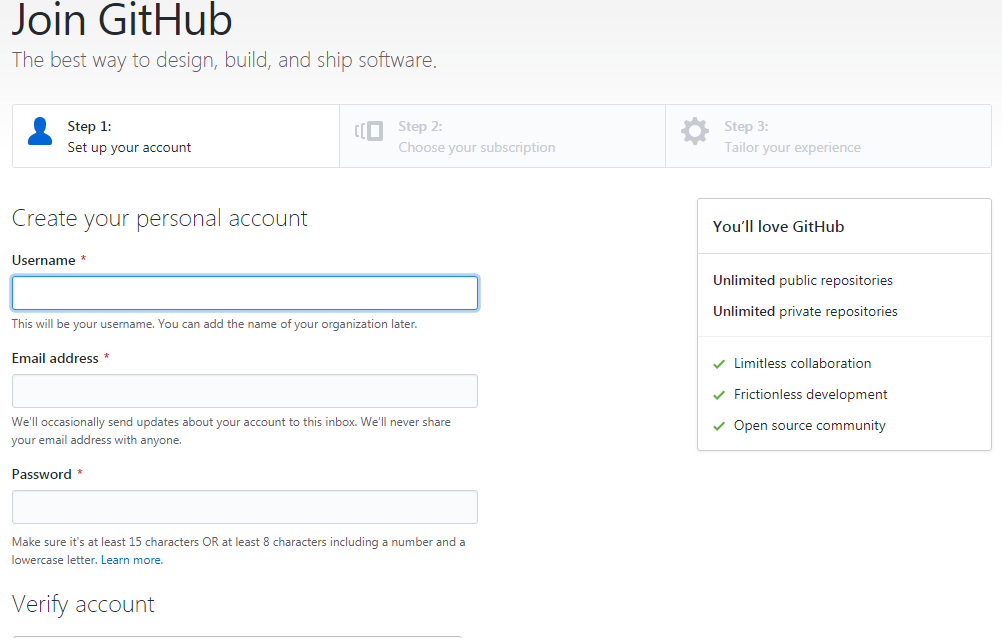
Version 5.0 is perfectly good there is no conflicts on that so you need to merge 5.0 on 4.7(Master branch) Git is open source free to download on the internet. If you have windows you need to download that one, if you have mac os it’s already exists.

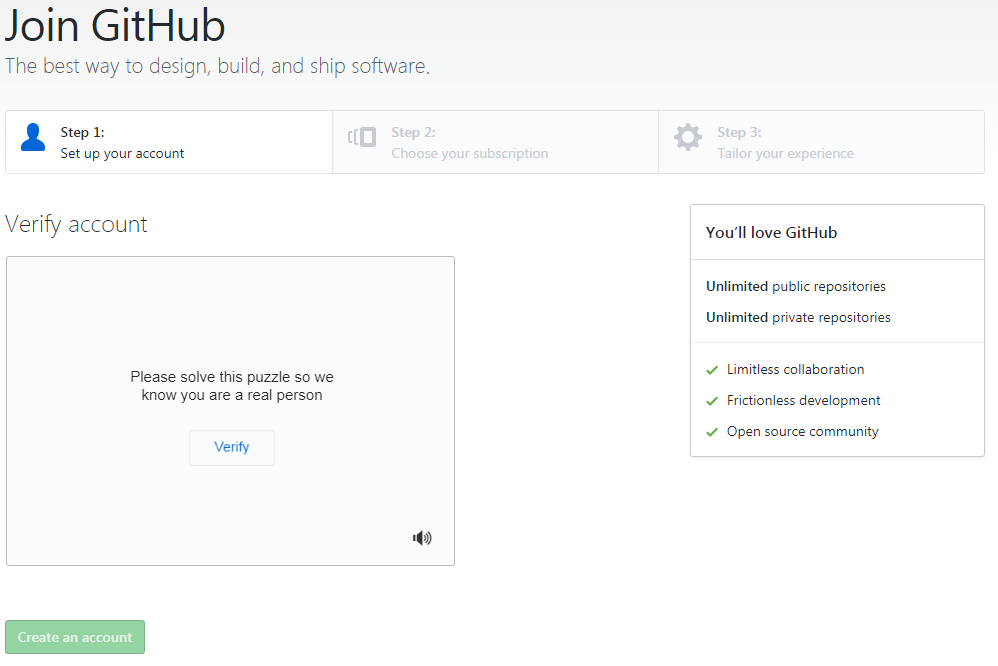
Signup for GitHub:

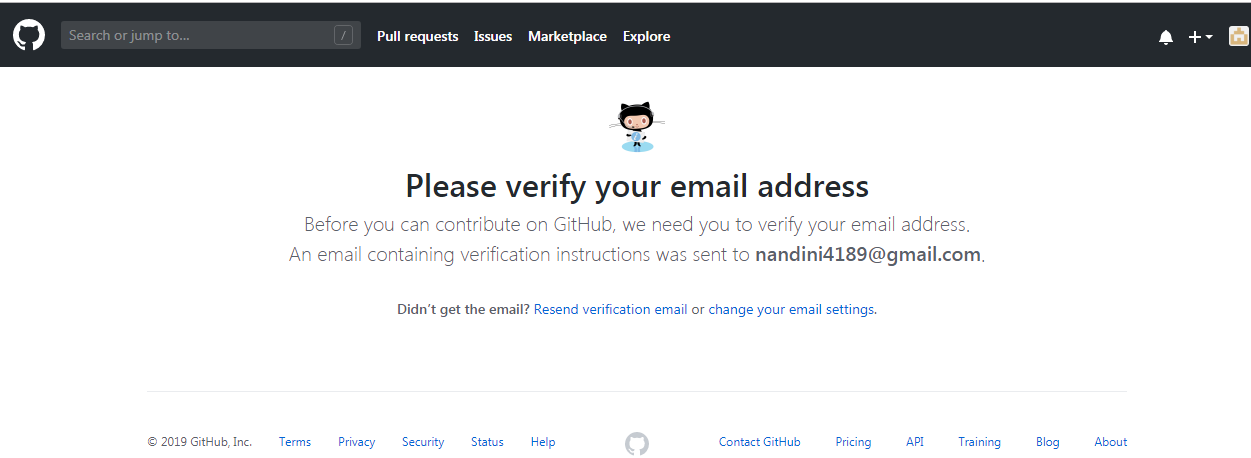
Go to <https://github.com>





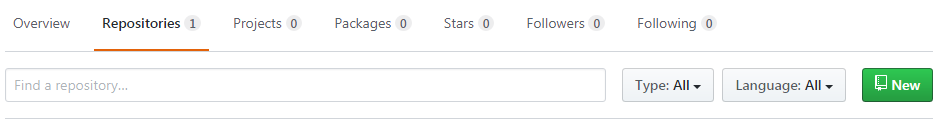


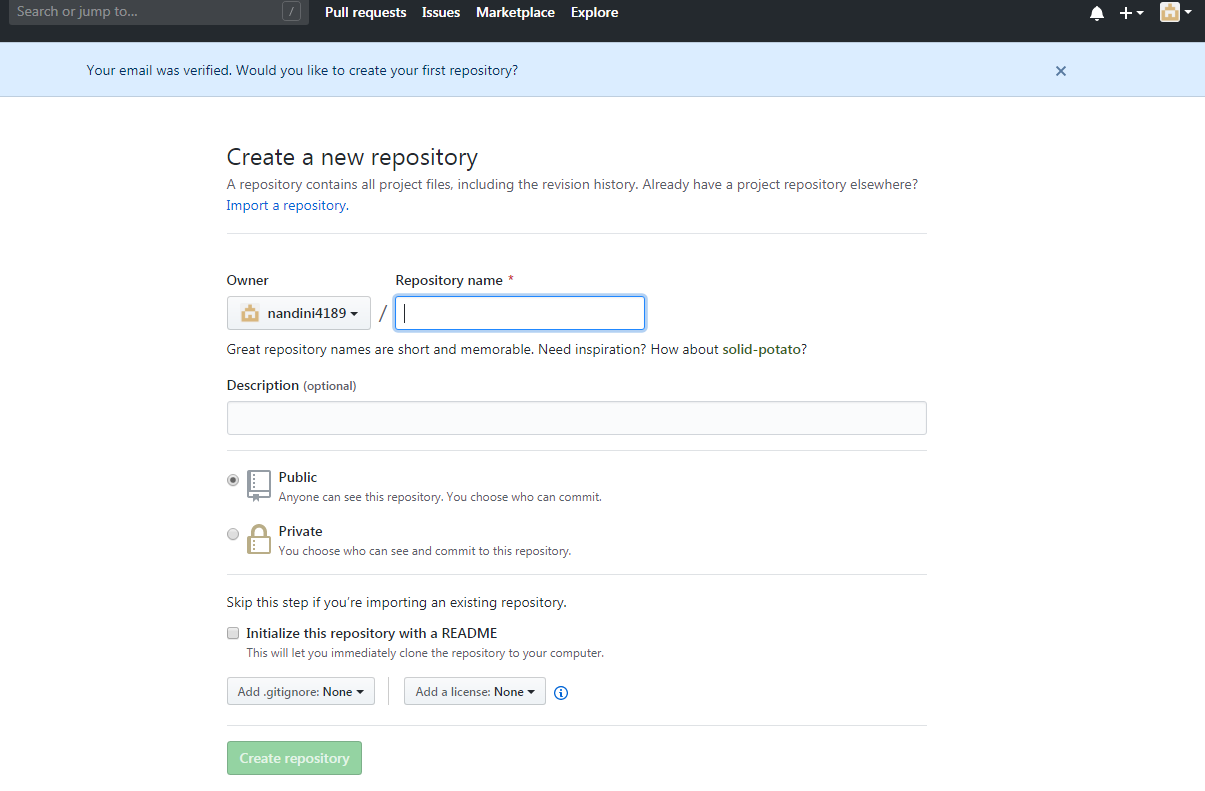




This is GitHub account all the stuff is here if we want to work with GitHub first we need to create one Repository by default there is no repositories on Github.

For creating new repository:

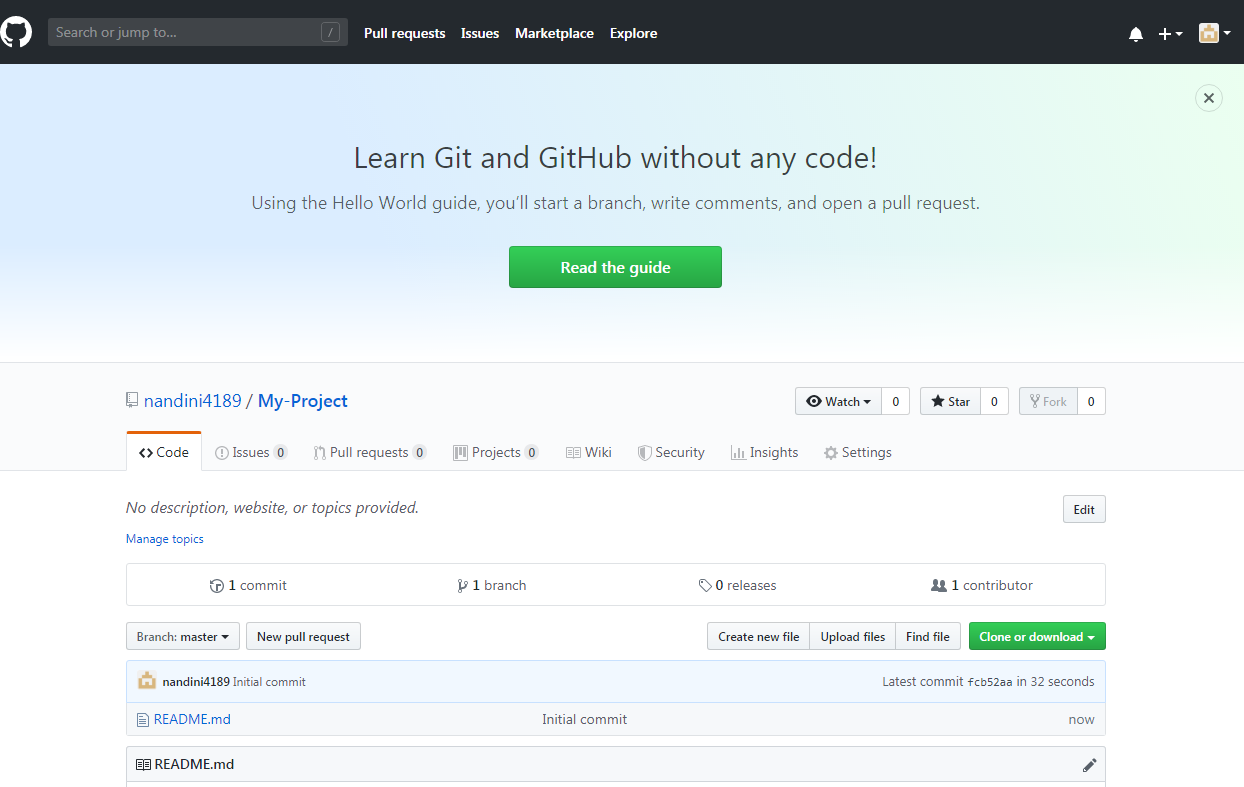




Here we can give a name of the repository and

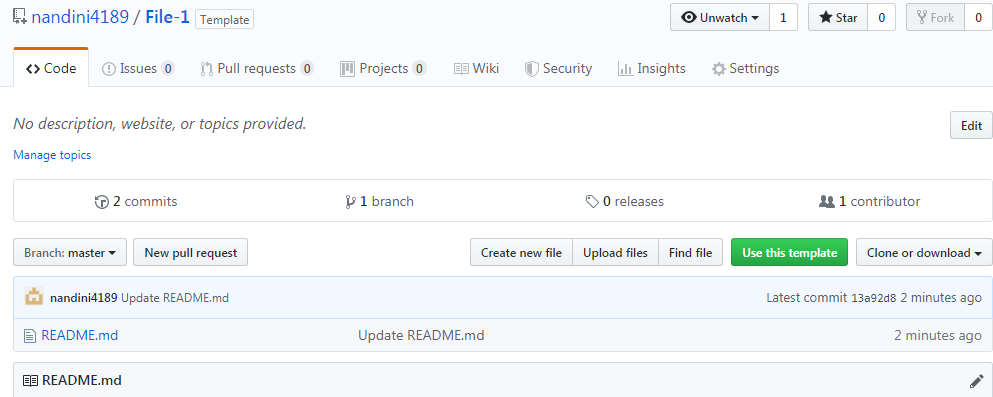
Description is optional, and goes for public, if you go for private it’s paid one.

And select initialize the repository with a readme after that click on create repository then it will create a new repository.

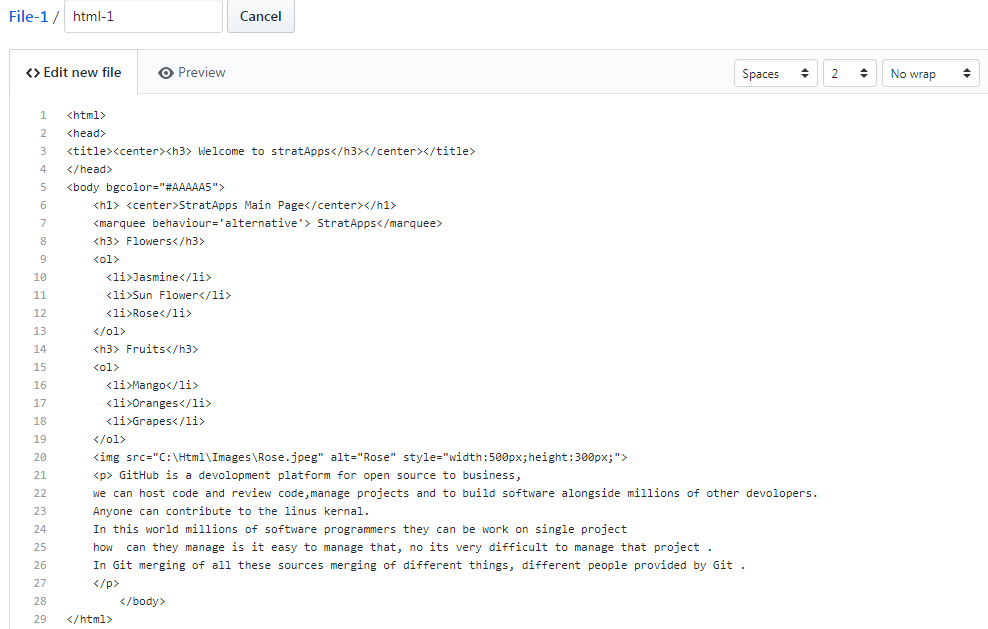


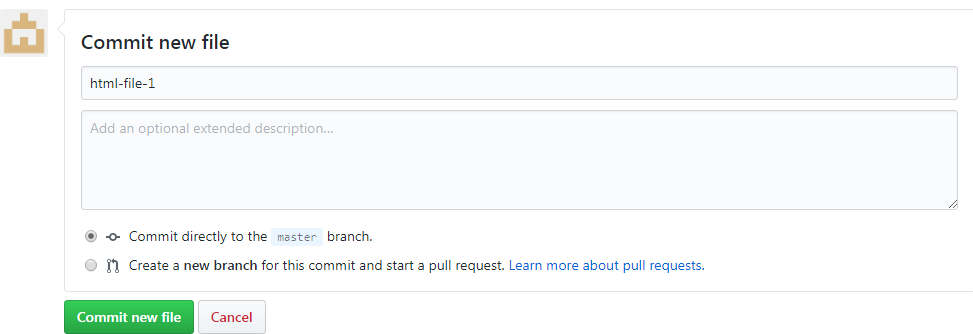
Here we have shown comment, branch, contributor..

Here the repository name as file-1

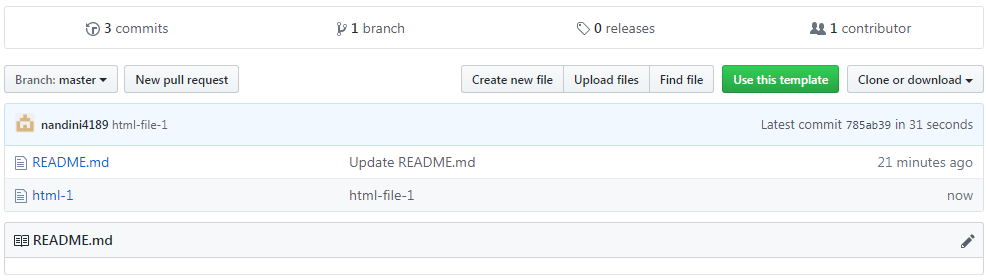


Create new file and we can write java file,c-sharp,php file,c file..whatever you want that file ,create the name of the file and add code on that

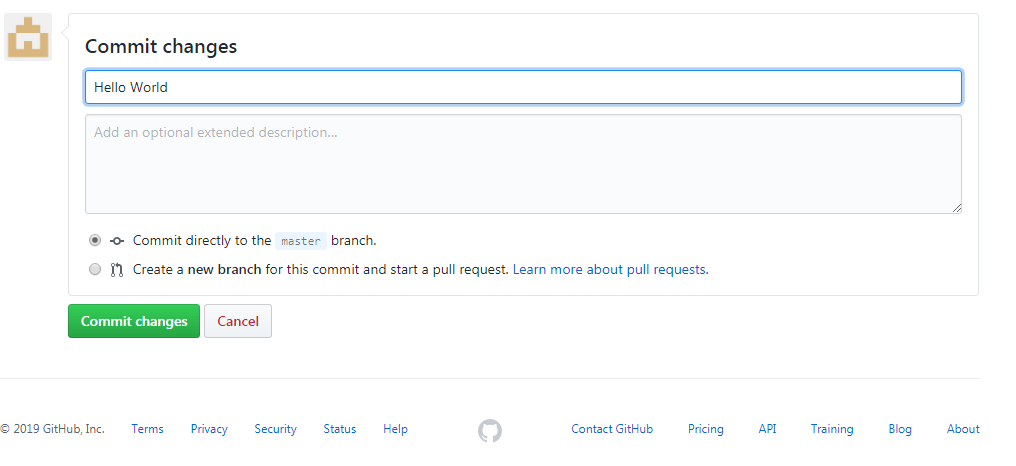


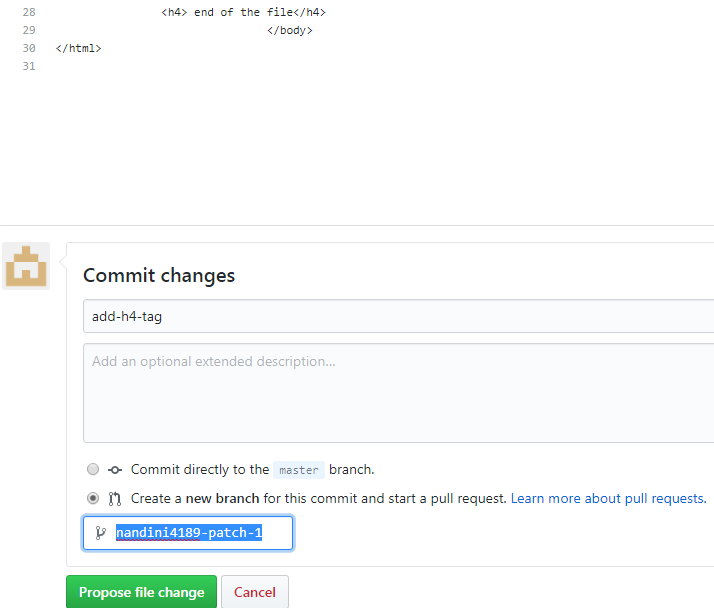


Here html-file-1 as file name and select the commit directly to the master branch.

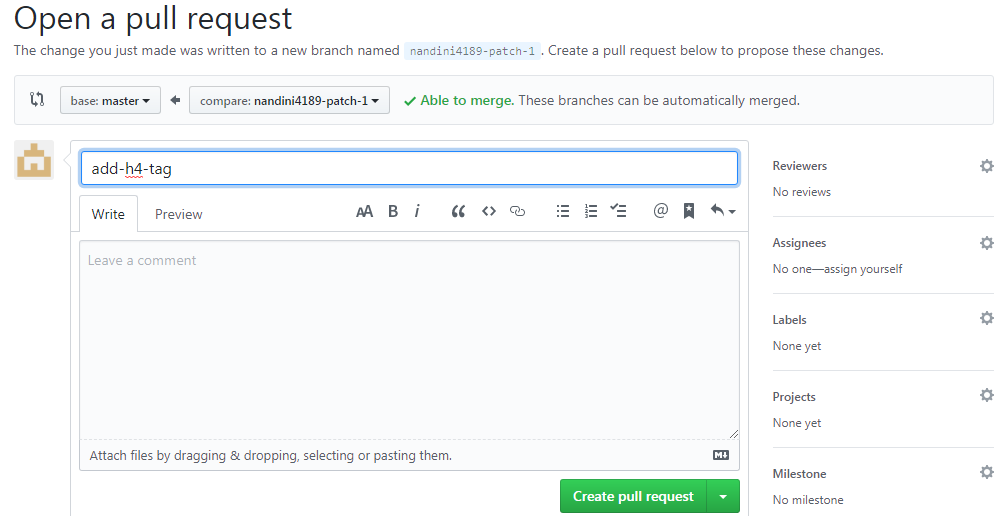


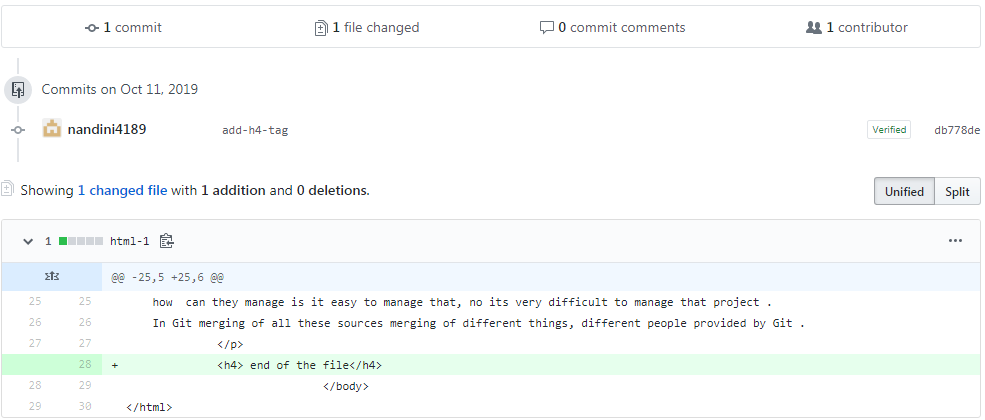
Here save the html-1 file, edit that file and add new features and save that file it will directly saves as master branch, if we want to create new branch also available. If we select new branch then new feature, old one also shown on the new branch. If we open old one only older code will be shown.



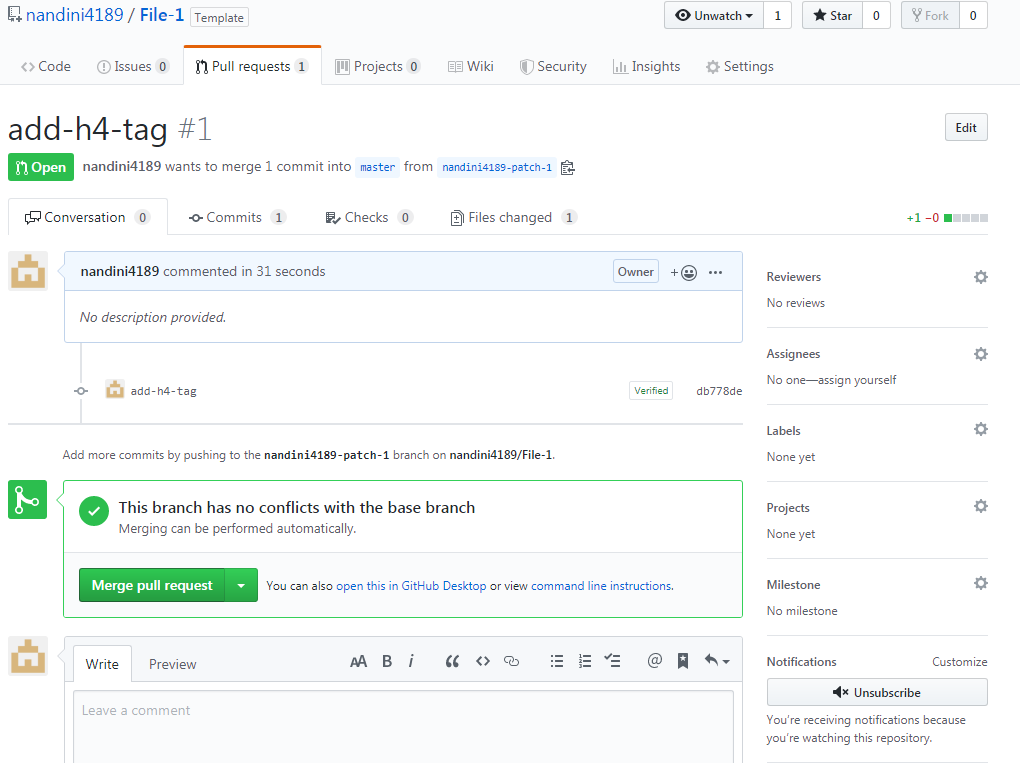


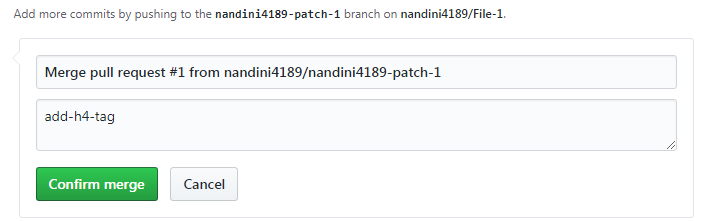
Here click on propose file change then it will go



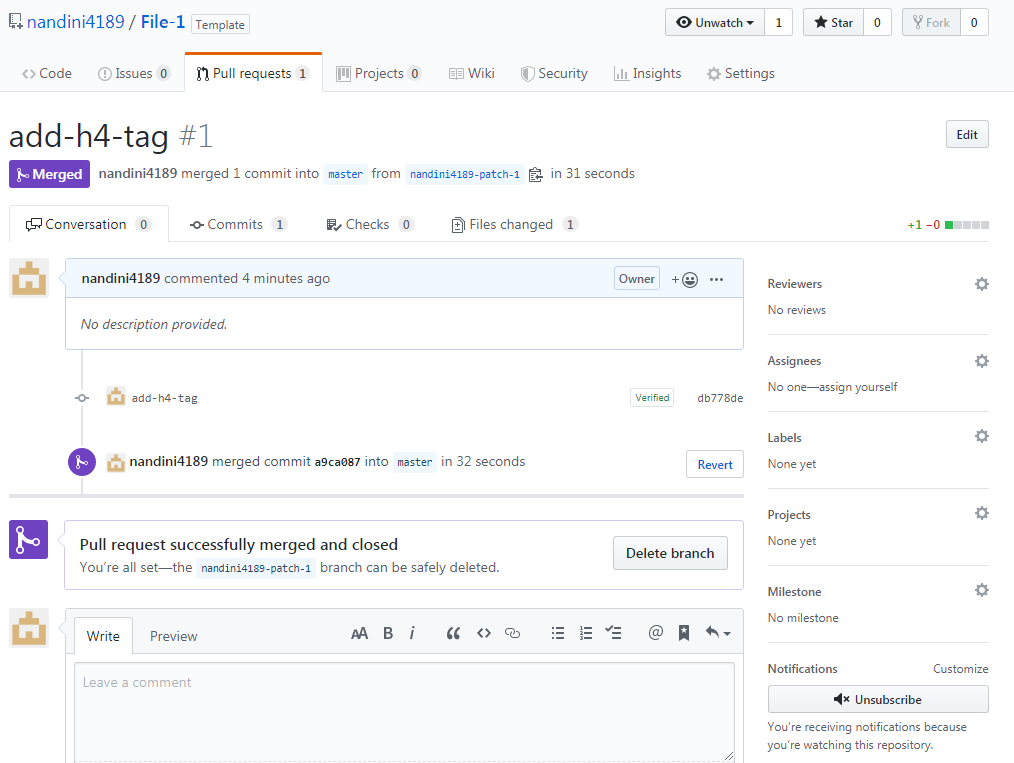


Green color identifies as adding one new tag.

We have master branch and new branch we need to merge these two files here Create Pull Requests and then it will show one msg as this branch has no conflicts with the base branch.



If we click on confirm merge then



The color will change and it will shows as pull request successfully merged and closed.